

Con-nect

Con-form Benutzerhandbuch

Version 3.4.2

November 2016

Dieses Dokument gilt für Con-nect ab Version 3.4.2.

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

Copyright © 1985-2016 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, USA, und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

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

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

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

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

Dokument-ID: CMF-USERGUIDE-342-20161125DE

Inhaltsverzeichnis

Con-form Benutzerhandbuch	vii
1 Con-form-Anweisungen	1
Syntax der Anweisungen	2
Syntax der Optionen	5
Anweisungen ignorieren	6
2 Ein einfaches Dokument erstellen	7
Allgemeine Informationen	8
Flattersatz	9
Blocksatz	10
Eine neue Zeile beginnen	13
Leerzeilen erzeugen	17
Kommentare in den Rohtext einfügen	20
3 Layout eines Dokuments	23
Bestandteile einer formatierten Seite	25
Anweisungen für das Dokument-Layout	27
Linker und rechter Rand	29
Heftrand	32
Leerzeilen am Seitenanfang	37
Leerzeilen am Seitenende	38
Titelzeile und Fußzeile	39
Mehrzeilige Titel- und Fußzeilen	51
Länge der Titel- und Fußzeilen	55
Seitennummerierung	58
Kopfzeilen	64
Fußnoten	69
Seitenlänge	77
Zeilenabstand und Schreibdichte	78
4 Text anordnen	85
Text ausrichten	87
Text einrücken	94
Seitenumbruch	103
Auf einer Seite verfügbare Zeilen	106
5 Text hervorheben	117
Fettdruck	119
Text unterstreichen	124
Kursivdruck	129
Text hochstellen	130
Text tiefstellen	131
Text überdrucken	132
Umschaltanweisungen	133
6 Silbentrennung	139
Silbentrennung in der formatierten Ausgabe	141
Im Rohtext getrennte Wörter	147

7	Tabulatoren und Rahmen	149
	Tabulatoren	150
	Rahmen	160
8	Große Dokumente	173
	Dokumente einbetten	175
	Einstellungen speichern und wiederherstellen	179
	Kapitelüberschriften und Inhaltsverzeichnis	181
	Indexeinträge und Index	202
9	Variablen	205
	Eigene Variablen definieren	206
	Arithmetische Berechnungen	213
	Textfunktionen	219
	Dezimalzahlen	222
	Das Variablenzeichen	227
	Systemvariablen	229
10	Bedingte und wiederholte Verarbeitung	233
	Bedingte Verarbeitung	235
	Wiederholte Verarbeitung	239
	Eine Bedingung definieren	242
	Eine Konsequenz definieren	245
	Logische Operatoren	246
11	Makros	249
	Allgemeine Informationen	250
	Ein Makro definieren	251
	Ein Makro aufrufen	253
	Lokale Makrovariablen	254
12	Steueranweisungen für die Ausgabe	259
	Die auszugebenden Seiten definieren	260
	Die Papierkassette definieren	262
	Leerzeichen	264
	Zeichen umsetzen	267
	Die Textrichtung ändern	276
	Fehlerbehebung	278
13	Ein Dokument formatieren	279
	Der Befehl FORMATIEREN	280
	Druckerprofile	280
	Formatierprofile	281
14	Spezielle Zeichen	283
	@	285
	Apostroph (')	285
	Dollar (\$)	285
	Kaufmännisches Und (&)	286
	Komma (,)	287
	Leerzeichen	287
	Nummernzeichen (#)	287

Punkt (.)	289
Schrägstrich (/)	289
Semikolon (;)	289
Zirkumflex (^)	289
15 Serienbriefe	291
Das Statement COMPOSE	292
Den Serienbrief testen	294
16 Veraltete Anweisungen	297
.>> - Sprungsymbol	299
.GO - Sprung	299
.LO - Kleinschreibung	300
.OP LOC - Umschaltverriegelungszeichen	300
.OP SHI - Umschaltzeichen für Groß-/Kleinschreibung	302
.PN - Seitenzahl	302
.TY - Text auf Bildschirm ausgeben	302
.UP - Großschreibung	303

Con-form Benutzerhandbuch

Con-form ist ein Textformatiersystem, das automatisch mit Con-nect installiert wird.

Zur Texterfassung können Sie den Con-nect-Editor benutzen (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Textverarbeitung*).

Diese Dokumentation bietet eine schrittweise Einführung in Con-form. Sie beginnt mit einfachen Anweisungen zum Formatieren von Text und geht dann zur Beschreibung der komplexeren Funktionen über.

Viele Con-form-Anweisungen werden anhand von Beispielen erklärt. Die Beispiele zeigen jeweils den zu formatierenden Text mit den Con-form-Anweisungen sowie die formatierte Ausgabe des Textes. In der Regel bezeichnet der Begriff "Ausgabe" die gedruckte Version des formatierten Dokuments.

Diese Dokumentation ist in die folgenden Abschnitte unterteilt:

Con-form-Anweisungen	Beschreibt, wie Sie Con-form-Anweisungen in einem Text angeben. Erklärt außerdem, wie die Erkennung der Con-form-Anweisungen unterdrückt werden kann.
Ein einfaches Dokument erstellen	Stellt die wichtigsten Anweisungen zum Erzeugen eines einfachen Dokuments vor.
Layout eines Dokuments	Beschreibt, wie das Format eines Dokuments gestaltet werden kann.
Text anordnen	Beschreibt Funktionen zum Anordnen von Text wie z.B. zentrieren, rechtsbündig anordnen, Abschnitt einrücken oder Seitenumbruch definieren.
Text hervorheben	Beschreibt Funktionen wie Fettdruck und Unterstreichung. Erklärt außerdem die Anwendung von Umschaltanweisungen, mit denen Sie z.B. Kursivdruck definieren können.
Silbentrennung	Beschreibt, wie die Silbentrennung für Englisch, Deutsch oder Französisch eingeschaltet wird.
Tabulatoren und Rahmen	Beschreibt, wie Sie Tabulatoren setzen und Text einrahmen.
Große Dokumente	Beschreibt, wie aus mehreren Dokumenten ein einziges Ausgabedokument erstellt wird (dies wird als "Einbetten" bezeichnet). Erklärt außerdem, wie Inhaltsverzeichnis und Index erstellt werden.
Variablen	Beschreibt das Arbeiten mit Variablen.
Bedingte und wiederholte Verarbeitung	Beschreibt, wie Anweisungsfolgen definiert werden, die nur dann ausgeführt werden, wenn das Ergebnis einer Bedingung wahr ist.
Makros	Beschreibt den Einsatz von Makros.
Steueranweisungen für die Ausgabe	Beschreibt, wie definiert wird, welche Seiten gedruckt werden und welche Papierkassette benutzt wird. Außerdem wird erklärt, wie mehrere aufeinander folgende Leerzeichen durch ein Leerzeichen ersetzt werden und wie Zeichen umgesetzt werden.
Ein Dokument formatieren	Beschreibt, wie ein Dokument formatiert, ein Druckerprofil benutzt und ein Formatierprofil definiert wird.
Spezielle Zeichen	Beschreibt die Zeichen, die in Con-form eine besondere Bedeutung haben.
Serienbriefe	Beschreibt, wie ein Formbrief angelegt wird, der Variablen enthält, die mit Daten aus einer Adabas-Datei ersetzt werden sollen. Dabei wird auch gezeigt, wie Sie testen können, ob Ihr Formbrief das gewünschte Ergebnis liefert, bevor die Variablen mit Daten aus der Datenbank ersetzt werden.
Veraltete Anweisungen	Listet alle Con-form-Anweisungen, die nur noch aus Gründen der Kompatibilität mit älteren Con-form-Versionen verfügbar sind. Es wird jedoch empfohlen, diese Anweisungen nicht zu verwenden.

Nachstehend finden Sie Folgendes:

- **Liste aller Anweisungen**
- **Liste aller Optionen**

Liste aller Anweisungen

Die folgende Liste aller Con-form-Anweisungen ist alphabetisch sortiert.

- **.** (comment)** - Kommentar
- **.>> (label)** - Sprungsymbol
- **.AN (and)** - Und - logischer Operator
- **.BF (boldface)** - Fettdruck
- **.BM (bottom margin)** - Leerzeilen nach der Fußzeile
- **.BP (bold print)** - Fettdruck
- **.BR (line break)** - Zeilenumbruch
- **.BT (bottom title for all pages)** - Fußzeile für alle Seiten
- **.BX (box)** - Rahmen zeichnen
- **.CB (compress blanks)** - Leerzeichen unterdrücken
- **.CE (center text)** - Text zentrieren
- **.CH (chapter)** - Kapitel
- **.CO (comment)** - Kommentar
- **.CS (character spacing)** - Schreibdichte
- **.CV (compute variable)** - Variable berechnen
- **.DU (dump workspace)** - Speicherauszug drucken
- **.EB (bottom title for even pages)** - Fußzeile für gerade Seiten
- **.EC (end-of-line character)** - Zeilenende-Zeichen
- **.EF (end of file)** - Ende der Datei
- **.EI (end If-condition)** - Ende der bedingten Verarbeitung
- **.EL (else)** - Alternative definieren
- **.EM (embed)** - einbetten
- **.EP (end of processing)** - Ende der Verarbeitung
- **.ET (top title for even pages)** - Titelzeile für gerade Seiten
- **.EW (end While-loop)** - Schleifenende
- **.FI (filling)** - Zeilenausgleich
- **.FM (footer margin)** - Leerzeilen zwischen Text und Fußzeile
- **.FN (footnote)** - Fußnote
- **.FS (footer space)** - Zeilenreservierung für mehrzeilige Fußzeile
- **.GO (go to specified label)** - Sprung

- **.HC (header clear) - Kopfzeile löschen**
- **.HL (header lines) - Kopfzeile**
- **.HM (header margin) - Leerzeilen zwischen Titelzeile und Text**
- **.HP (hyphen connection) - Trennung aufheben**
- **.HS (header space) - Zeilenreservierung für mehrzeilige Titelzeile**
- **.IC (ignore instructions) - Anweisungen ignorieren**
- **.IF (start If-condition) - wenn - Bedingung**
- **.IL (insert lines) - Zeilen einfügen**
- **.IP (insert pages) - Seiten einfügen**
- **.IX (index) - Indexeintrag**
- **.JU (justification) - Randausgleich**
- **.LL (line length) - Zeilenlänge**
- **.LM (left margin) - linker Rand**
- **.LO (lower-case) - Kleinschreibung ein/aus**
- **.LS (line spacing) - Zeilenabstand**
- **.MA (start macro) - Beginn eines Makros**
- **.ME (end macro) - Ende eines Makros**
- **.MX (exit from macro) - Makro abbrechen**
- **.NL (need lines) - benötigte Zeilen**
- **.NP (new page) - neue Seite**
- **.OB (bottom title for odd pages) - Fußzeile für ungerade Seiten**
- **.OF (offsetting) - Einzug mehrerer Zeilen**
- **.OP (option) - Option**
- **.OR (or) - Oder - logischer Operator**
- **.OT (top title for odd pages) - Titelzeile für ungerade Seiten**
- **.PH (print header) - Kopfzeile im Text drucken**
- **.PL (page length) - Seitenlänge**
- **.PM (page-numbering mode) - Form der Seitennummerierung**
- **.PN (page number) - Seitenzahl**
- **.PS (paragraph start) - Absatz**
- **.PT (put to table of contents) - Eintrag in das Inhaltsverzeichnis**
- **.RA (right-adjustment) - rechtsbündige Textausgabe**
- **.RM (right margin) - rechter Rand**

- **.RS (restore context) - Kontext wiederherstellen**
- **.SA (save context) - Kontext speichern**
- **.SB (skip blanks) - Leerzeichen am Zeilenanfang ignorieren**
- **.SC (set chapter number) - Kapitelnummer setzen**
- **.SL (skip lines) - Zeilen überspringen**
- **.SU (substitution) - Ersetzung ein/aus**
- **.SV (set variable) - Variable definieren**
- **.TB (tab stops) - Tabulator setzen**
- **.TH (then) - Blockkonsequenz definieren**
- **.TI (temporary indentation) - Einzug einer Zeile**
- **.TM (top margin) - Leerzeilen über der Titelzeile**
- **.TO (text orientation) - Textrichtung ändern**
- **.TR (translate character to character) - Zeichen umsetzen**
- **.TS (translate character to string) - Zeichen in Zeichenkette umsetzen**
- **.TT (top title for all pages) - Titelzeile für alle Seiten**
- **.TY (type information) - Text auf Bildschirm ausgeben**
- **.UL (underline) - unterstreichen**
- **.UP (upper-case) - Großschreibung ein/aus**
- **.US (underscore) - unterstreichen**
- **.WH (start While-loop) - Schleifenkopf**
- **.WX (exit from While-loop) - Schleife verlassen**

Liste aller Optionen

Die folgende Liste aller Con-form-Optionen ist alphabetisch sortiert.

- **BIN (paper bin) - Papierkassette wählen**
- **BRN (suppress line breaks) - kein Zeilenumbruch bei .NL**
- **BXH (horizontal box line) - waagerechte Linie für Rahmen**
- **BXV (vertical box line) - senkrechte Linie für Rahmen**
- **CHA (chapter numbering) - Kapitelnummerierung**
- **CHI (chapter indentation) - Einrückung im Inhaltsverzeichnis**
- **CHL (chapter levels) - Kapitelstufen im Inhaltsverzeichnis**
- **CSE (instruction separator character) - Anweisungstrennzeichen ändern**
- **DAS (characters after decimal) - Stellen nach Dezimalzeichen**

- **DEC (decimal character) - Dezimalzeichen definieren**
- **ECH (echo instructions) - Anweisungen ausgeben ein/aus**
- **EMN (embed nobreak) - kein Zeilenumbruch beim Einbetten**
- **END (end-of-line character) - Zeilenende-Zeichen**
- **ESC (escape character) - Umschaltzeichen definieren**
- **GEV (gutter for even pages) - Heftrand für gerade Seiten**
- **GOD (gutter for odd pages) - Heftrand für ungerade Seiten**
- **GUT (gutter for all pages) - Heftrand für alle Seiten**
- **HYA (characters after the hyphen) - Zeichen nach Trennung**
- **HYB (characters before the hyphen) - Zeichen vor Trennung**
- **HYP (hyphenation) - Silbentrennung**
- **LOC (shiftlock character) - Umschaltverriegelung**
- **NPG (number of pages to output) - Anzahl der auszugebenden Seiten**
- **PAG (page number) - aktuelle oder nächste Seitenzahl**
- **PGF (page formatting) - Seitenformatierung**
- **PNS (page-number character) - Zeichen der Seitennummerierung**
- **PTC (switch .PT on/off) - Anweisung .PT ein-/ausschalten**
- **REM (remainder of division) - Rest bei Division mit .CV**
- **RND (round result of arithmetic calculation) - Ergebnis von .CV runden**
- **ROM (Roman page-numbering) - Groß-/Kleinbuchstaben bei römischen Seitenzahlen**
- **SHI (shift character) - Umschaltzeichen für Groß-/Kleinschreibung**
- **SSF (single sheet feeder) - Papierkassette wechseln**
- **STA (start output at specific page) - Ausgabe ab einer bestimmten Seite**
- **STO (stop output at specific page) - Ausgabe bis zu einer bestimmten Seite**
- **TRI (thousands separator character) - Tausendertrennzeichen definieren**
- **ULB (underscore blanks) - Leerzeichen unterstreichen**
- **VSG (variable character) - Variablenzeichen definieren**

1 Con-form-Anweisungen

▪ Syntax der Anweisungen	2
▪ Syntax der Optionen	5
▪ Anweisungen ignorieren	6

Dieses Kapitel beschreibt, wie Con-form-Anweisungen im zu formatierenden Text angegeben werden. Es erklärt außerdem, wie Anweisungen ignoriert werden können.

Die folgenden Themen werden behandelt:

Syntax der Anweisungen

Die Beschreibung einer Con-form-Anweisung beginnt mit einem Syntaxdiagramm, das alle Möglichkeiten zur Angabe der Anweisung zeigt. Beispiel:

```
.RA zahl  
.RA ON  
.RA OFF
```

Die Anweisung ist in Großbuchstaben angegeben. Die Parameter, für die Sie eine Zahl, ein Zeichen oder einen Text eingeben müssen, sind in Kleinschreibung und kursiv dargestellt.



Anmerkung: Alle Anweisungen in den Syntaxdiagrammen sind in Großbuchstaben dargestellt. Sie können die Anweisungen und Parameter jedoch in jeder beliebigen Kombination von Groß- und Kleinbuchstaben eingeben.

In den Syntaxdiagrammen kommen folgende Symbole vor:

[]	Elemente in eckigen Klammern sind optional.
{ }	Von den Elementen in geschweiften Klammern muss genau eines (d.h. nur eines) angegeben werden.
...	Ein vor dem Auslassungszeichen stehendes Element kann wiederholt werden.

Der Punkt

Jede Con-form-Anweisung beginnt mit einem Punkt (.). Beispiel:

```
.BR
```

Eine Anweisung beginnt in der ersten Spalte einer Zeile. Es ist aber auch möglich, Text und Anweisungen in einer Zeile zu mischen. Dies wird bei der Beschreibung des Anweisungstrennzeichens erklärt.

Wenn Sie mehr als einen Punkt angeben, wird die Eingabe nicht als Con-form-Anweisung interpretiert.

Die Parameter

Bei bestimmten Anweisungen müssen Sie einen oder mehrere Parameter angeben.

Wenn eine Anweisung einen Parameter hat, muss zwischen der Anweisung und dem Parameter ein Leerzeichen stehen. Beispiel:

```
.FI ON
```

Wenn eine Anweisung mehrere Parameter hat, müssen Sie zwischen den Parametern jeweils ein Komma (,) eingeben. Beispiel:

```
.BX 10,55
```

Das Anweisungstrennzeichen

Wenn Sie mehrere Anweisungen in einer Zeile eingeben möchten, müssen Sie zwischen den Anweisungen ein bestimmtes Trennzeichen eingeben. Anfänglich ist dies ein Semikolon. Beispiel:

```
.FI ON;.JU ON
```

Sie können auch Text und Anweisungen in einer Zeile mischen. Ein Anweisungstrennzeichen gefolgt von einem Punkt (;.) zeigt immer an, dass eine Anweisung folgt (zwischen dem Anweisungstrennzeichen und dem Punkt darf *kein* Leerzeichen stehen). Beispiel:

```
Dies ist Text;.IL;Dies ist noch ein Text;.IL;Und noch ein Text
```

```
.CE ON;Überschrift zentrieren;.CE OFF;.IL;Dies steht unter der Überschrift.
```



Vorsicht: Die Anweisung nach dem Anweisungstrennzeichen wird nur erkannt, wenn sie direkt hinter dem Anweisungstrennzeichen steht. Wenn Sie zwischen dem Anweisungstrennzeichen und dem Punkt ein Leerzeichen eingeben, werden die folgenden Zeichen nicht als Anweisung interpretiert.

.OP CSE - Anweisungstrennzeichen ändern

```
.OP CSE=zeichen
```

Das Semikolon ist das vorgegebene Anweisungstrennzeichen. Sie können ein anderes Zeichen dafür definieren.

Beispiel: Um den Doppelpunkt (:) als Anweisungstrennzeichen zu definieren, geben Sie die folgende Anweisung an:

```
.OP CSE=:
```

Syntax der Optionen

.OP - Option

```
.OP schlüsselwort=wert[,schlüsselwort=wert]...
```

Optionen sind spezielle Anweisungen, die bestimmte Eigenschaften von Con-form festlegen. Falls erforderlich, können Sie den vorgegebenen Wert einer Option ändern.

Zwischen .OP und dem Schlüsselwort müssen Sie ein Leerzeichen eingeben. Zwischen dem Schlüsselwort und dem Wert muss ein Gleichheitszeichen stehen. Weder vor noch nach dem Gleichheitszeichen darf ein Leerzeichen stehen.

Das folgende Beispiel zeigt, wie eine Option angegeben wird. In diesem Fall ist CSE das Schlüsselwort und der Doppelpunkt (:) der Wert.

```
.OP CSE=:
```

Sie können mehrere Optionen in einer Anweisung angeben. Geben Sie dazu zwischen den Optionen jeweils ein Komma ein. Das folgende Beispiel definiert drei verschiedene Optionen:

```
.OP CSE=: ,END=% ,ESC=/'
```

Wenn ein Zeichen bereits für einen bestimmten Zweck benutzt wird (wie z.B. das Komma, das dazu dient, die Optionen der Anweisung .OP voneinander zu trennen), müssen Sie das Zeichen in Apostrophe einschließen. Beispiel:

```
.OP DEC=' , '
```

Die Optionen sind in den Abschnitten dieser Dokumentation beschrieben, die das jeweilige Thema behandeln.

Anweisungen ignorieren

.IC - Anweisungen ignorieren

```
.IC ON  
.IC OFF
```

Da Con-form einen Punkt in der ersten Spalte einer Zeile als Beginn einer Anweisung interpretiert, dürfen Sie dieses Zeichen nicht in der ersten Spalte des Textes eingeben. Sie können jedoch unterdrücken, dass Anweisungen als solche erkannt werden. Auf diese Weise können Sie den Punkt in der ersten Spalte benutzen und Con-form-Anweisungen im formatierten Text ausgeben.

Um Anweisungen zu ignorieren, geben Sie die folgende Anweisung ein:

```
.IC ON
```

Alle dieser Anweisung folgenden Zeichen erscheinen in der formatierten Ausgabe. Mit der folgenden Anweisung machen Sie diesen Schritt wieder rückgängig, so dass die Con-form-Anweisungen wieder ausgeführt werden. Diese Anweisung muss in der *ersten* Spalte einer Zeile beginnen:

```
.IC OFF
```

2 Ein einfaches Dokument erstellen

- Allgemeine Informationen 8
- Flattersatz 9
- Blocksatz 10
- Eine neue Zeile beginnen 13
- Leerzeilen erzeugen 17
- Kommentare in den Rohtext einfügen 20

Dieses Kapitel beschreibt die wichtigsten Anweisungen, die Sie für ein einfaches Dokument brauchen. Dabei wird nur auf den "Rumpf" des Dokuments eingegangen, d.h. auf den Text selbst.

Die folgenden Themen werden behandelt:

Zum Text außerhalb des Dokumentrumpfs gehören die Titel- und Fußzeilen, Kopfzeilen, Fußnoten, Indexeinträge sowie Einträge für das Inhaltsverzeichnis. Diese Textteile sind in späteren Abschnitten beschrieben.

Allgemeine Informationen

Wenn Sie einen Text schreiben, der keine Con-form-Anweisungen enthält, sehen die unformatierte und die formatierte Version des Textes gleich aus. Der Grund hierfür ist, dass der Zeilenausgleich anfänglich ausgeschaltet ist (siehe [.FI - Zeilenausgleich](#)).

Der einzige Unterschied zwischen der formatierten und der unformatierten Version sind drei Leerzeilen am Anfang jeder Seite. Diese Leerzeilen entstehen auf Grund der Anfangswerte `.HS 1` (siehe [.HS - Zeilenreservierung für mehrzeilige Titelzeile](#)) und `.HM 2` (siehe [.HM - Leerzeilen zwischen Titelzeile und Text](#)). Beim Drucken des formatierten Dokuments wird ein linker Rand von 10 Leerzeichen vor jeder Zeile ausgegeben. Dieser Rand entsteht auf Grund des Anfangswertes `.OP GUT=10` (siehe [.OP GUT - Heftrand für alle Seiten](#)). Wenn Sie nichts anderes angeben, gelten die Anfangswerte von Con-form für das Layout Ihres Dokuments.

Flattersatz

.FI - Zeilenausgleich

```
.FI ON
.FI OFF
```

Zeilenausgleich ist der Vorgang, bei dem Con-form eine Zeile zwischen den definierten Rändern mit der maximal möglichen Anzahl von Wörtern füllt.

Die Vorteile des Zeilenausgleichs sind:

- Sie können Ihren Text editieren ohne auf die Zeilenlänge zu achten; d.h. Sie können zusätzliche Wörter in der Mitte eines Abschnitts als extra Textzeile einfügen.
- Sie können den linken und rechten Rand ändern; Ihr Text wird automatisch dem neuen Rand entsprechend angeordnet.

Der Zeilenausgleich ist anfänglich ausgeschaltet (.FI OFF). In diesem Fall wird der für den rechten Rand definierte Wert nicht berücksichtigt, und der formatierte Text sieht genauso aus wie der unformatierte Text.

Wenn Sie Ihren Text zwischen dem definierten linken und rechten Rand so anordnen möchten, dass der rechte Rand als Flatterrand gedruckt wird, müssen Sie den Zeilenausgleich einschalten. Dabei ist zu beachten, dass der Zeilenausgleich mit dem Randausgleich (.JU) zusammenhängt (der anfänglich ausgeschaltet ist), d.h. der Randausgleich muss ausgeschaltet werden:

```
.FI ON;.JU OFF
```

Selbst für ein einfaches Dokument ist zu empfehlen, den linken und rechten Rand zu definieren (siehe [Linker und rechter Rand](#)) und den Zeilenausgleich einzuschalten. Beispiel:

```
.LM 0;.RM 60
.FI ON;.JU OFF
```



Anmerkung: Wenn Sie den Zeilenausgleich einschalten, aber den linken und rechten Rand nicht definieren, gelten die Anfangswerte .LM 0 (linker Rand) und .RM 72 (rechter Rand).

Blocksatz

.JU - Randausgleich

```
.JU ON  
.JU OFF
```

Wenn Zeilenausgleich *und* Randausgleich eingeschaltet sind, werden zwischen den Wörtern Leerzeichen so eingefügt, dass der Text zwischen dem linken und rechten Rand bündig ausgerichtet wird. In diesem Fall wird die formatierte Ausgabe nicht mehr mit Flatterrand sondern im Blocksatz gedruckt.

Zeilen, denen ein Zeilenumbruch folgt (z.B. die letzte Zeile eines Abschnitts), werden nicht ausgerichtet.

Der Randausgleich ist anfänglich eingeschaltet (.JU ON). Das bedeutet, dass der Randausgleich automatisch aktiviert wird, wenn Sie am Anfang Ihres Dokuments .FI ON angeben. Es wird jedoch empfohlen, immer beide Anweisungen anzugeben, wenn Sie den Randausgleich einschalten möchten:

```
.FI ON; .JU ON
```

Wenn .JU für den Text definiert wurde, gilt dies nicht automatisch für die Fußnoten (siehe [.FN - Fußnote](#)).

Beispiel

Im folgenden Rohtext werden die Anweisungen `.FI` und `.JU` zum Zeilenausgleich und Randausgleich benutzt. Mit den Anweisungen `.LM` und `.RM` werden der linke und der rechte Rand definiert (siehe *Linker und rechter Rand*). Außerdem enthält das Beispiel die Anweisung `.SL` zum Einfügen von Leerzeilen (siehe *.SL - Zeilen überspringen*).

Rohtext

```
.LM 0;.RM 60
Der Zeilenausgleich ist standardmäßig ausgeschaltet (.FI OFF), und der
Randausgleich ist eingeschaltet (.JU ON). Bei ausgeschaltetem
Zeilenausgleich werden Wörter nicht über eine Zeile hinaus verschoben.
Falls ein linker und ein rechter Rand angegeben sind, werden diese nicht
berücksichtigt, und der formatierte Text sieht genauso aus wie der
unformatierte.
.SL 1
.FI ON
Nun ist der Zeilenausgleich eingeschaltet (.FI ON). Da der Randausgleich
standardmäßig eingeschaltet ist (.JU ON), werden Leerzeichen zwischen den
Wörtern eingefügt, um einen geraden linken und rechten Rand, d.h. Blocksatz,
zu erreichen. Der Randausgleich ist nur bei eingeschaltetem Zeilenausgleich
möglich.
.SL 1
.JU OFF
Nun ist der Randausgleich ausgeschaltet (.JU OFF). Das bedeutet, dass der
rechte Rand ein Flatterrand bleibt. Da aber der Zeilenausgleich immer noch
eingeschaltet ist (.FI ON), werden kurze Zeilen mit Wörtern aus anderen
Zeilen aufgefüllt. Die Angaben für den linken und rechten Rand werden
berücksichtigt.
```

Formatierte Ausgabe

Der Zeilenausgleich ist standardmäßig ausgeschaltet (.FI OFF), und der Randausgleich ist eingeschaltet (.JU ON). Bei ausgeschaltetem Zeilenausgleich werden Wörter nicht über eine Zeile hinaus verschoben. Falls ein linker und ein rechter Rand angegeben sind, werden diese nicht berücksichtigt, und der formatierte Text sieht genauso aus wie der unformatierte.

Nun ist der Zeilenausgleich eingeschaltet (.FI ON). Da der Randausgleich standardmäßig eingeschaltet ist (.JU ON), werden Leerzeichen zwischen den Wörtern eingefügt, um einen geraden linken und rechten Rand, d.h. Blocksatz, zu erreichen. Der Randausgleich ist nur bei eingeschaltetem Zeilenausgleich möglich.

Nun ist der Randausgleich ausgeschaltet (.JU OFF). Das

bedeutet, dass der rechte Rand ein Flatterrand bleibt. Da aber der Zeilenausgleich immer noch eingeschaltet ist (.FI ON), werden kurze Zeilen mit Wörtern aus anderen Zeilen aufgefüllt. Die Angaben für den linken und rechten Rand werden berücksichtigt.

Eine neue Zeile beginnen

Wenn der Zeilenausgleich eingeschaltet ist, wird der Text so angeordnet, dass jede Zeile mit möglichst vielen Wörtern gefüllt wird. Um im formatierten Text eine neue Zeile zu beginnen, können Sie Unterbrechungen des Zeilenausgleichs angeben. Dazu gibt es mehrere Möglichkeiten.

.BR - Zeilenumbruch

```
.BR
```

Um eine neue Zeile zu beginnen, können Sie die Anweisung `.BR` benutzen.

.EC - Zeilenende-Zeichen ändern

```
.EC zeichen
```

Der Anfangswert für das Zeilenende-Zeichen ist das Dollarzeichen. Der diesem Zeichen folgende Text erscheint in der nächsten Zeile.

Das Zeilenende-Zeichen muss im Rohtext als letztes Zeichen in einer Zeile eingegeben werden. Andernfalls wird es als normaler Text interpretiert.

Um z.B. das Prozentzeichen als Zeilenende-Zeichen zu definieren, geben Sie die folgende Anweisung ein:

```
.EC %
```

.OP END - Zeilenende-Zeichen ändern

```
.OP END=zeichen
```

Anstelle der Anweisung `.EC` können Sie auch `.OP END` benutzen. Um das Prozentzeichen als neues Zeilenende-Zeichen zu definieren, geben Sie ein:

```
.OP END=%
```

Leerzeichen

Ein oder mehrere Leerzeichen am Anfang einer Zeile bewirken einen Zeilenumbruch. Trotzdem erscheinen die Leerzeichen auch im formatierten Text am Anfang der Zeile.

Es erfolgt kein Zeilenumbruch, wenn die folgende Anweisung angegeben wurde:

```
.SB ON
```

Siehe auch: [.CB - Leerzeichen unterdrücken](#).

Leerzeilen

Sie können Leerzeilen in den Rohtext einfügen. In der formatierten Ausgabe beginnt der Text nach den Leerzeilen mit einer neuen Zeile.

Anweisungen, die einen Zeilenumbruch bewirken

Wenn Sie eine der folgenden Anweisungen verwenden, beginnt der auf die Anweisung folgende Text in einer neuen Zeile.

```
.BF  
.BX  
.CE  
.CH  
.EF  
.EM  
.FI  
.IL  
.IP  
.LM  
.NL  
.NP  
.OF  
.PH  
.PS  
.RA  
.SL  
.TI  
.UL
```

Beispiel

In diesem Beispiel wird die Anweisung `.BR` sowie das Zeilenende-Zeichen benutzt, um im formatierten Text eine neue Zeile zu beginnen. Anfänglich ist das Dollarzeichen (\$) das Zeilenende-Zeichen. Im Beispiel wird das Zeilenende-Zeichen mit der Anweisung `.EC` geändert. Außerdem zeigt das Beispiel, wie eine neue Zeile begonnen wird: entweder mit Hilfe von Leerzeichen am Anfang einer Zeile oder einfach durch Leerzeilen im Rohtext.

Rohtext

```
.LM 0;.RM 65
.FI ON
```

Wenn Text in einer neuen Zeile beginnen soll, können Sie das Zeilenende-Zeichen im Rohtext angeben. Standardmäßig ist dies das Zeichen \$. Der auf das Dollarzeichen folgende Text wird in der nächsten Zeile erscheinen. Das Dollarzeichen muss am Ende einer Zeile stehen.\$
Wenn \$ nicht am Zeilenende steht, wird es als normaler Text interpretiert.

```
.SL 1
```

Sie können das Zeilende-Zeichen ändern. Benutzen Sie dazu eine der folgenden Anweisungen: `.OP END=zeichen` oder `.EC zeichen`.

```
.SL 1
.EC &
```

Das kaufmännische Und (&) ist nun das Zeilenende-Zeichen.&
Das heißt, das Dollarzeichen wird nun als Text interpretiert, wenn es als letztes Zeichen in einer Zeile steht:\$

```
.BR
```

Um eine neue Zeile zu beginnen, können Sie auch die Anweisung `.BR` verwenden.

Ein Zeilenumbruch erfolgt auch, wenn im Text Leerzeilen vorkommen.
Auch ein Leerzeichen am Zeilenanfang bewirkt einen Zeilenumbruch. Das Leerzeichen am Zeilenanfang bleibt im formatierten Text bestehen.

Formatierte Ausgabe

Wenn Text in einer neuen Zeile beginnen soll, können Sie das Zeilenende-Zeichen im Rohtext angeben. Standardmäßig ist dies das Zeichen \$. Der auf das Dollarzeichen folgende Text wird in der nächsten Zeile erscheinen. Das Dollarzeichen muss am Ende einer Zeile stehen.
Wenn \$ nicht am Zeilenende steht, wird es als normaler Text interpretiert.

Sie können das Zeilende-Zeichen ändern. Benutzen Sie dazu eine der folgenden Anweisungen: `.OP END=zeichen` oder `.EC zeichen`.

Das kaufmännische Und (&) ist nun das Zeilenende-Zeichen.
Das heißt, das Dollarzeichen wird nun als Text interpretiert, wenn es als letztes Zeichen in einer Zeile steht:\$

Um eine neue Zeile zu beginnen, können Sie auch die Anweisung `.BR` verwenden.

Ein Zeilenumbruch erfolgt auch, wenn im Text Leerzeilen vorkommen.

Auch ein Leerzeichen am Zeilenanfang bewirkt einen Zeilenumbruch. Das Leerzeichen am Zeilenanfang bleibt im formatierten Text bestehen.

Leerzeilen erzeugen

Sie können im Rohtext entweder Zeilen leer lassen oder mit Hilfe der folgenden Anweisungen Leerzeilen explizit angeben. Wie Leerzeilen am Anfang und Ende einer Seite definiert werden, ist in den folgenden Abschnitten beschrieben: *.TM - Leerzeilen über der Titelzeile* und *.BM - Leerzeilen nach der Fußzeile*.

Die Anweisungen `.SL` und `.IL` sind ähnlich. Es wird empfohlen, zum Einfügen von Leerzeilen am Seitenanfang die Anweisung `.IL` zu benutzen.



Anmerkung: Sie können auch die Anweisung `.PS` verwenden, um eine Leerzeile im formatierten Text auszugeben (siehe *.PS - Absatz*).

.SL - Zeilen überspringen

```
.SL zahl  
.SL
```

Wenn Sie zwei Abschnitte trennen möchten, wird empfohlen, die Anweisung `.SL` zu benutzen. Das Trennen zweier Abschnitte wird entweder durch Leerzeilen oder durch einen Seitenumbruch erreicht. Die Anweisung `.SL` fügt keine Leerzeilen am Anfang einer Seite ein.

Beispiel: Um drei Leerzeilen einzufügen, geben Sie die folgende Anweisung ein:

```
.SL 3
```

Wenn Sie keine Zahl angeben, wird eine Leerzeile erzeugt.

Wenn die angegebene Zahl größer ist als die Anzahl der restlichen Zeilen auf der Seite, werden die darüber hinausgehenden Zeilen ignoriert. Beispiel: Wenn Sie `.SL 10` angeben, aber nur noch zwei Zeilen auf die Seite passen, werden die übrigen acht Zeilen ignoriert.

.IL - Zeilen einfügen

```
.IL zahl  
.IL
```

Wenn Sie am Anfang einer Seite Leerzeilen angeben möchten (z.B. um Platz frei zu lassen für eine Abbildung, die später der gedruckten Ausgabe hinzugefügt werden soll), wird empfohlen, die Anweisung `.IL` zu benutzen (und nicht die Anweisung `.SL`).

Außerdem müssen Sie auch die Anweisung `.NL` oder `.NP` angeben. Beispiel: Um 10 Leerzeilen am Seitenanfang einzufügen, können Sie eine der folgenden Kombinationen von Anweisungen benutzen:

```
.NL 10;.IL 10
```

```
.NP;.IL 10
```

Wenn bei der Anweisung `.IL` eine Zahl angegeben wird, die größer ist als die Anzahl der restlichen Zeilen auf der Seite, werden die über die Seite hinausgehenden Zeilen am Anfang der folgenden Seite eingefügt.

Wenn Sie bei der Anweisung `.IL` keine Zahl angeben, wird eine Zeile eingefügt.

Sie können auch `.IL 0` angeben. Damit wird ein Zeilenumbruch erzeugt, d.h. diese Anweisung entspricht der Anweisung `.BR`. In diesem Fall werden keine Leerzeilen erzeugt.

Beispiel

Dieses Beispiel zeigt die verschiedenen Möglichkeiten zum Erzeugen von Leerzeilen.

Rohtext

```
.LM 0;.RM 60
```

```
.FI ON;.JU ON
```

Sie können entweder Zeilen im Rohtext leer lassen oder eine Con-form-Anweisung angeben.

Die Anweisungen `.SL` und `.IL` sind ähnlich. Es wird jedoch empfohlen, zum Einfügen von Leerzeilen am Seitenanfang die Anweisung `.IL` zu benutzen.

```
.SL 1
```

Wenn Sie zwei Abschnitte trennen möchten, wird die Anweisung `.SL` empfohlen. Die beiden Abschnitte werden entweder durch Leerzeilen oder durch einen Seitenumbruch getrennt. Im Gegensatz zur Anweisung `.IL` erzeugt `.SL` keine Leerzeilen am Seitenanfang.

Formatierte Ausgabe

Sie können entweder Zeilen im Rohtext leer lassen oder eine Con-form-Anweisung angeben.

Die Anweisungen `.SL` und `.IL` sind ähnlich. Es wird jedoch empfohlen, zum Einfügen von Leerzeilen am Seitenanfang die Anweisung `.IL` zu benutzen.

Wenn Sie zwei Abschnitte trennen möchten, wird die Anweisung `.SL` empfohlen. Die beiden Abschnitte werden entweder durch Leerzeilen oder durch einen Seitenumbruch getrennt. Im Gegensatz zur Anweisung `.IL` erzeugt `.SL` keine Leerzeilen am Seitenanfang.

Kommentare in den Rohtext einfügen

Ein Kommentar ist eine Information im Rohtext, die in der formatierten Ausgabe nicht erscheint. Ein Kommentar wird entweder mit der Anweisung `**` oder mit `.CO` eingeleitet.

Ein Kommentar kann beliebige Zeichen enthalten und darf höchstens eine Zeile lang sein. Das Zeilenende-Zeichen (anfänglich das Dollarzeichen) wird nicht interpretiert, wenn es in einem Kommentar vorkommt.

Ein Kommentar bewirkt keinen Zeilenumbruch.

Die Anweisungen `**` und `.CO` sind nicht ganz identisch. Wenn die Textzeile vollständig ignoriert werden soll, wird empfohlen, die Anweisung `**` zu verwenden.

`` - Alle Anweisungen nach dem Kommentar ignorieren**

```
** text
```

Im Gegensatz zur Anweisung `.CO` wird der Rest der Zeile, in der `**` eingegeben wurde, ignoriert; d.h. eine Anweisung nach der Anweisung `**` wird nicht ausgeführt.

Das Anweisungstrennzeichen (Anfangswert: Semikolon) wird ignoriert, d.h. es kann im Kommentar benutzt werden. Beispiel:

```
** Dies ist ein Kommentar; er erscheint nicht in der formatierten Ausgabe
```

`.CO` - Alle Anweisungen nach dem Kommentar ausführen

```
.CO text
```

Die Anweisung `.CO` kann in einer Zeile benutzt werden, die mehrere Anweisungen enthält; d.h. vor und nach `.CO` können weitere Anweisungen stehen (getrennt durch das Anweisungstrennzeichen). Der Anfangswert für das Anweisungstrennzeichen ist das Semikolon.

Wenn Sie das Anweisungstrennzeichen in einem Kommentar benutzen möchten, müssen Sie es wiederholen. Andernfalls erscheint der nach dem Anweisungstrennzeichen stehende Text in der formatierten Ausgabe. Beispiel:


```
.FI ON;.CO Dies ist Kommentar;; die nächste Anweisung wird ausgeführt;.JU ON
```

Beispiel

Dieses Beispiel zeigt die beiden Anweisungen, mit denen Kommentare im Rohtext eingegeben werden können.

Rohtext

```
.LM 0;.RM 60
.FI ON
Sie können Kommentare in Ihren Rohtext einfügen, die in der formatierten
Ausgabe nicht erscheinen.
.** Dies ist Kommentar; Anweisungen in dieser Zeile werden ignoriert;.LM 10
.** Wenn Sie mehrere Kommentarzeilen eingeben möchten, muss jede Zeile mit
.** der Anweisung .** oder .CO beginnen.
Ein Kommentar bewirkt keine Unterbrechung des Zeilenausgleichs.
.SL 1
.CO Noch ein Kommentar;; die folgende Anweisung wird ausgeführt;.LM 10
Nun ist der neue linke Rand wirksam, der nach dem Kommentar definiert wurde,
der mit der Anweisung .CO eingeleitet wurde.
```

Formatierte Ausgabe

Sie können Kommentare in Ihren Rohtext einfügen, die in der formatierten Ausgabe nicht erscheinen. Ein Kommentar bewirkt keine Unterbrechung des Zeilenausgleichs.

Nun ist der neue linke Rand wirksam, der nach dem Kommentar definiert wurde, der mit der Anweisung .CO eingeleitet wurde.

3

Layout eines Dokuments

- Bestandteile einer formatierten Seite 25
- Anweisungen für das Dokument-Layout 27
- Linker und rechter Rand 29
- Heftrand 32
- Leerzeilen am Seitenanfang 37
- Leerzeilen am Seitenende 38
- Titelzeile und Fußzeile 39
- Mehrzeilige Titel- und Fußzeilen 51
- Länge der Titel- und Fußzeilen 55
- Seitennummerierung 58
- Kopfzeilen 64
- Fußnoten 69
- Seitenlänge 77
- Zeilenabstand und Schreibdichte 78

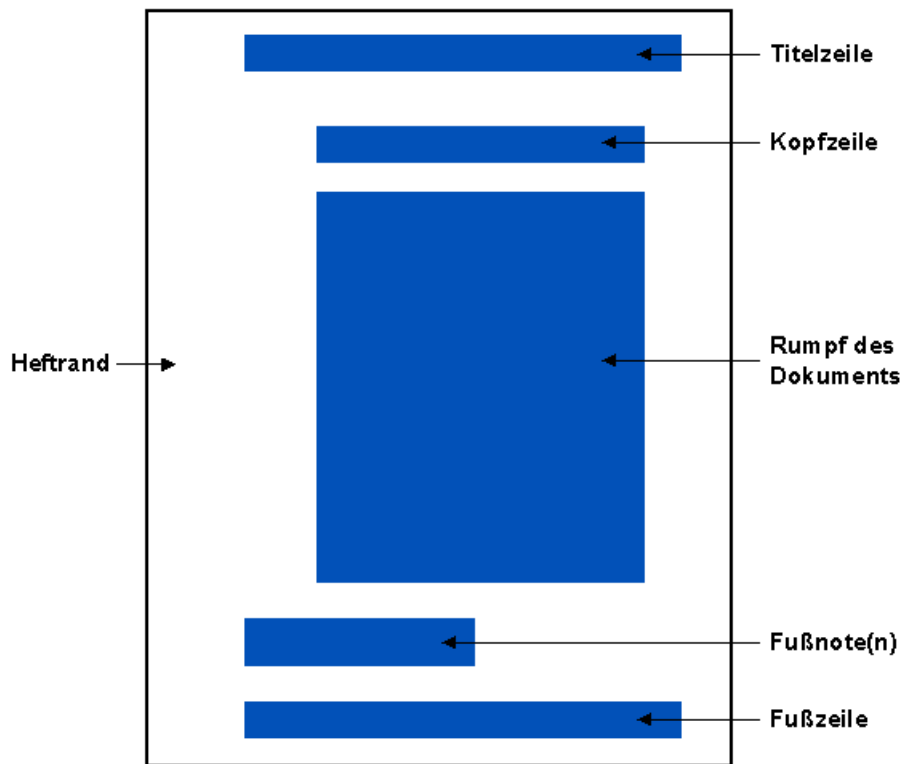
Dieses Kapitel beschreibt, wie Sie Ihr Dokument individuell gestalten. Wenn Sie nichts anderes angeben, gelten die Anfangswerte von Con-form für das Layout des Dokuments.

Die folgenden Themen werden behandelt:

Sie können ein Formatierprofil anlegen, das *vor* dem Rohtext verarbeitet wird (siehe [Formatierprofile](#)). Das Formatierprofil kann z.B. Ihre eigenen Anfangswerte für das Seiten-Layout enthalten.

Bestandteile einer formatierten Seite

Die folgende Abbildung zeigt die verschiedenen Bestandteile einer formatierten Seite:



Heftrand

Der Heftrand ist ein zusätzlicher Rand, der für die Lochung des Papiers vorgesehen ist. Sie können für gerade und ungerade Seiten verschiedene Werte für den Heftrand angeben.

Titelzeile

Die Titelzeile erscheint ganz oben auf jeder Seite. Wenn Sie z.B. ein Buch schreiben, können Sie den Buchtitel und den Kapitelnamen oben auf jeder Seite ausgeben. Sie können auch die aktuelle Seitenzahl ausdrucken. Es ist möglich, verschiedene Titelzeilen für gerade und ungerade Seiten zu definieren.

Kopfzeile

Die Kopfzeile erscheint oben auf jeder Seite zwischen der Titelzeile und dem Rumpf des Dokuments. Sie können die Kopfzeile auch im Dokumentrumpf ausgeben lassen.

Rumpf des Dokuments

Der Dokumentrumpf enthält den eigentlichen Text.

Fußnote(n)

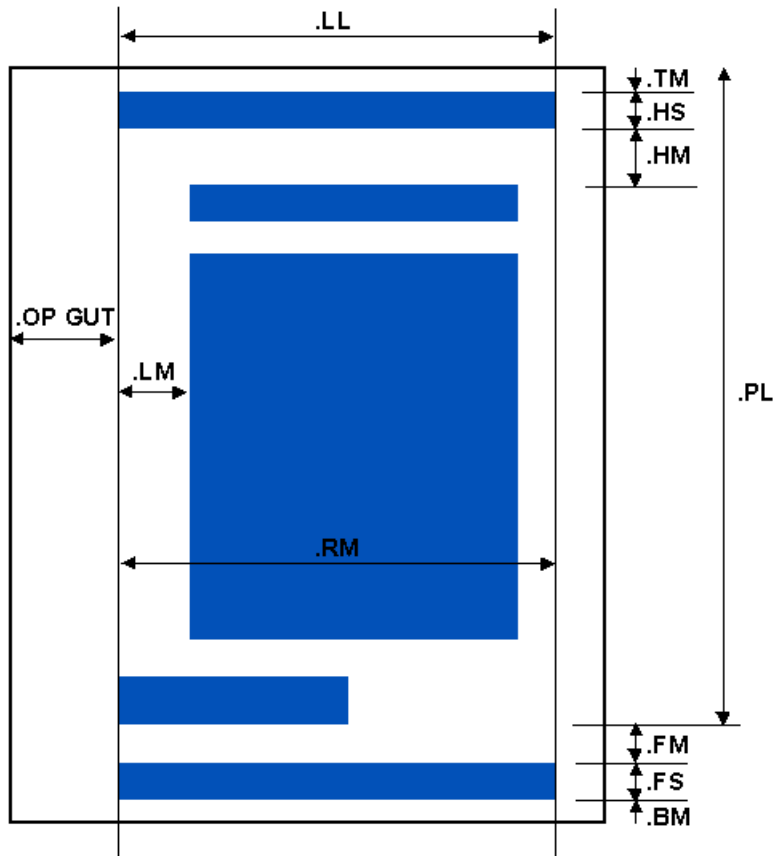
Eine Fußnote steht zwischen dem Dokumentrumpf und der Fußzeile. Sie bezieht sich auf eine Textstelle auf der aktuellen Seite.

Fußzeile

Die Fußzeile erscheint am unteren Rand jeder Seite. Sie können z.B. die aktuelle Seitenzahl unten auf jeder Seite ausgeben. Für die Fußzeile können Sie auch definieren, dass die Seitenzahl der nächsten Seite ausgegeben wird (was in der Titelzeile nicht möglich ist). Für gerade und ungerade Seiten sind unterschiedliche Fußzeilen möglich.

Anweisungen für das Dokument-Layout

Die folgende Abbildung zeigt die Anweisungen, mit denen das Layout einer Seite definiert wird.



.LM und .RM

Diese Anweisungen definieren die Spalten für den linken und den rechten Rand.

.OP GUT

Diese Anweisung definiert die Anzahl der Spalten, die auf allen Seiten als Heftrand reserviert werden. Der Heftrand kann für gerade und ungerade Seiten unterschiedlich definiert werden (.OP GOD und .OP GEV).

.TM, .HM, .BM und .FM

Diese Anweisungen definieren die Anzahl von Leerzeilen zwischen den verschiedenen Teilen der Seite.

.HS und .FS

Diese Anweisungen definieren, wie viele Zeilen für die Titelzeile und die Fußzeile reserviert werden sollen.

.LL

Diese Anweisung definiert die Zeilenlänge (d.h. die Anzahl von Zeichen) für die Titel- und Fußzeile.

.PL

Diese Anweisung definiert die Seitenlänge, d.h. die Anzahl der Zeilen pro Seite insgesamt (einschließlich Leerzeilen über der Titelzeile, Titelzeile, Kopfzeile, Text und Fußnoten).

Linker und rechter Rand

Mit dem linken und rechten Rand wird die Zeilenlänge für die Kopfzeile und den Dokumenttrumpf definiert. Die Titel- und Fußzeile sind davon nicht betroffen (siehe [.LL - Zeilenlänge](#)).

.LM - Linker Rand

```
.LM zahl  
.LM +zahl  
.LM -zahl  
.LM
```

Der Anfangswert für den linken Rand ist Spalte 0. Dies ist der kleinstmögliche absolute Wert für den linken Rand. Der Text beginnt immer nach der Spalte, die als linker Rand definiert wurde. Beispiel: Wenn Sie `.LM 10` angeben, beginnt der Text in Spalte 11. Die Variable `$IN` enthält den aktuellen Wert für den linken Rand (siehe [Veränderbare Systemvariablen](#)).

Es gibt mehrere Möglichkeiten, den linken Rand mit der Anweisung `.LM` zu ändern. Die folgenden Beispiele zeigen die verschiedenen Möglichkeiten.

Sie geben eine Spaltennummer für den linken Rand an, z.B.:

```
.LM 10
```

Wenn der aktuelle linke Rand 10 ist und Sie ihn um 5 Spalten nach rechts (d.h. auf Spalte 15) verschieben möchten, geben Sie an:

```
.LM +5
```

Wenn der aktuelle linke Rand 15 ist und Sie ihn um 10 Spalten nach links (d.h. auf Spalte 5) verschieben möchten, geben Sie an:

```
.LM -10
```

Wenn Sie die Anweisung ohne Parameter angeben, entspricht dies dem Anfangswert `.LM 0`.

.RM - Rechter Rand

```
.RM zahl  
.RM +zahl  
.RM -zahl  
.RM
```

Der Anfangswert für den rechten Rand ist Spalte 72; der Höchstwert ist 190. Der rechte Rand beginnt in der angegebenen Spalte. Beispiel: Wenn Sie Spalte 60 angeben, wird das letzte Zeichen der Zeile in Spalte 59 ausgegeben. Die Variable \$RM enthält den aktuellen Wert für den rechten Rand (siehe [Veränderbare Systemvariablen](#)).

Es gibt mehrere Möglichkeiten, den rechten Rand mit der Anweisung .RM zu ändern. Die folgenden Beispiele zeigen die verschiedenen Möglichkeiten.

Sie definieren eine bestimmte Spalte für den rechten Rand, z.B. Spalte 60:

```
.RM 60
```

Wenn Spalte 60 der aktuelle rechte Rand ist und Sie ihn um 5 Spalten nach rechts verschieben möchten (d.h. auf Spalte 65), geben Sie an:

```
.RM +5
```

Wenn der aktuelle rechte Rand in Spalte 65 beginnt und Sie ihn um 10 Spalten nach links verschieben möchten, so dass er in Spalte 55 beginnt, geben Sie an:

```
.RM -10
```

Wenn Sie keinen Parameter angeben, entspricht dies dem Anfangswert .RM 72.

Beispiel

Dieses Beispiel zeigt, wie mit den Anweisungen .LM und .RM der linke und der rechte Rand gesetzt und geändert werden kann.

Rohtext

```
.FI ON;.JU ON
Weder der linke noch der rechte Rand wurden für diesen Teil des Textes
definiert. Es gelten die Anfangswerte: .LM 0 für den linken Rand und .RM 72
für den rechten Rand.
.SL 1
.LM 10;.RM 60
Nun wurde Spalte 10 für den linken Rand und Spalte 60 für den rechten Rand
definiert.
.SL 1
.LM +5;.RM -5
Nun wurde der linke Rand um 5 Spalten nach rechts verschoben, d.h. auf
Spalte 15. Der rechte Rand wurde um 5 Spalten nach links verschoben; er
beginnt nun in Spalte 55.
.SL 1
.LM -5;.RM +5
Mit diesen Anweisungen wurden die Ränder wieder auf die vorherigen Werte
zurückgesetzt.
```

Formatierte Ausgabe

Weder der linke noch der rechte Rand wurden für diesen Teil des Textes definiert. Es gelten die Anfangswerte: .LM 0 für den linken Rand und .RM 72 für den rechten Rand.

Nun wurde Spalte 10 für den linken Rand und Spalte 60 für den rechten Rand definiert.

Nun wurde der linke Rand um 5 Spalten nach rechts verschoben, d.h. auf Spalte 15. Der rechte Rand wurde um 5 Spalten nach links verschoben; er beginnt nun in Spalte 55.

Mit diesen Anweisungen wurden die Ränder wieder auf die vorherigen Werte zurückgesetzt.

Heftrand

Um ausgedruckte Seiten abheften zu können, kann ein zusätzlicher Rand definiert werden, der als "Heftrand" bezeichnet wird.

In Con-form definieren Sie den Heftrand, indem Sie angeben, wie viele Leerzeichen vor dem *linken* Rand stehen sollen. Wenn Sie z.B. den Heftrand vergrößern, wird der gesamte Text nach rechts verschoben.

Der Anfangswert für den Heftrand ist 10 Spalten vor dem linken Rand. Dieser Wert gilt für alle Seiten (gerade und ungerade).



Anmerkung: Wenn es in Ihrer Umgebung möglich ist, die Schreibdichte zu ändern (siehe *.CS - Schreibdichte*), interpretiert Con-form die Anzahl von Leerzeichen in zehntel Zoll. In diesem Fall kann es sein, dass die Schreibdichte für den Heftrand von der Schreibdichte des Textes abweicht.

Außer dem Heftrand für alle Seiten gibt es auch die Möglichkeit, für gerade und ungerade Seiten verschiedene Werte für den Heftrand zu definieren. Dies ist sinnvoll für ein Dokument, das als Buch gebunden werden soll. In diesem Fall wird empfohlen, einen kleineren Wert für die geraden Seiten zu definieren, damit der Text nach links verschoben wird, und einen größeren Wert für die ungeraden Seiten, damit der Text nach rechts verschoben wird.

Der Heftrand bestimmt die Position des linken Randes auf der gedruckten Seite. Er ändert jedoch nicht die Spaltennummer des linken Randes (die mit *.LM* definiert wurde). Wenn für den linken Rand z.B. 10 definiert ist und Sie den Heftrand vergrößern, bleibt der Wert für den linken Rand unverändert.

Der Heftrand wird mit der Anweisung *.OP* definiert und erscheint nur in der gedruckten Version der formatierten Ausgabe. Wenn Sie den Text am Bildschirm formatieren, wird der Heftrand nicht angezeigt.

.OP GUT - Heftrand für alle Seiten

```
.OP GUT=zah1
```

Diese Anweisung gilt sowohl für gerade als auch für ungerade Seiten. Als *zah1* geben Sie an, wie viele Spalten vor dem linken Rand leer bleiben sollen.

Beispiel: Wenn vor dem linken Rand 15 Spalten leer bleiben sollen, geben Sie an:

```
.OP GUT=15
```

Diese Anweisung wird am Anfang der nächsten Seite wirksam.

.OP GEV - Heftrand für gerade Seiten

```
.OP GEV=zah1
```

Diese Anweisung gilt für die geraden Seiten. Dies sind normalerweise die linken Seiten eines Buches. Als *zah1* geben Sie an, wie viele Spalten vor dem linken Rand leer bleiben sollen.

Beispiel: Wenn vor dem linken Rand jeder geraden Seite 9 Zeichen leer bleiben sollen, geben Sie an:

```
.OP GEV=9
```

Diese Anweisung wird am Anfang der nächsten Seite wirksam.

.OP GOD - Heftrand für ungerade Seiten

```
.OP GOD=zah1
```

Diese Anweisung gilt für die ungeraden Seiten. Dies sind normalerweise die rechten Seiten eines Buches. Als *zah1* geben Sie an, wie viele Spalten vor dem linken Rand leer bleiben sollen.

Beispiel: Wenn vor dem linken Rand jeder ungeraden Seite 21 Spalten leer bleiben sollen, geben Sie an:

.OP GOD=21

Diese Anweisung wird am Anfang der nächsten Seite wirksam.

Beispiel

Dieses Beispiel zeigt, wie für gerade und ungerade Seiten unterschiedliche Hefränder definiert werden. Beachten Sie, dass die Position des linken Randes auf der gedruckten Seite durch den Heftrand bestimmt wird. Außerdem benutzt dieses Beispiel die Anweisung `.NP`, die einen Seitenbruch bewirkt (siehe *`.NP - Neue Seite`*).

Rohtext

```
.** Der Heftrand für ungerade Seiten ist 21:  
.OP GOD=21  
.** Der Heftrand für gerade Seiten ist 9:  
.OP GEV=9  
.LM 0;.RM 50  
.FI ON;.JU ON
```

Dieser Text soll auf einer Seite mit ungerader Seitenzahl gedruckt werden. Der Heftrand für ungerade Seiten beträgt 21 Leerzeichen vor dem linken Rand. Wenn Sie vorhaben, ein Buch zu schreiben, sollte der Heftrand für die ungeraden Seiten (dies sind normalerweise die rechten Seiten eines Buches) größer sein, damit der Text nach rechts verschoben wird.

```
.NP
```

Dieser Text soll auf einer Seite mit gerader Seitenzahl stehen. Der Heftrand für gerade Seiten beträgt 9 Leerzeichen vor dem linken Rand. Wenn Sie vorhaben, ein Buch zu schreiben, sollte der Heftrand für die geraden Seiten (dies sind normalerweise die linken Buchseiten) kleiner sein, so dass der Text nach links verschoben wird.

Formatierte Ausgabe

Dieser Text soll auf einer Seite mit ungerader Seitenzahl gedruckt werden. Der Heftrand für ungerade Seiten beträgt 21 Leerzeichen vor dem linken Rand. Wenn Sie vorhaben, ein Buch zu schreiben, sollte der Heftrand für die ungeraden Seiten (dies sind normalerweise die rechten Seiten eines Buches) größer sein, damit der Text nach rechts verschoben wird.

Dieser Text soll auf einer Seite mit gerader Seitenzahl stehen. Der Heftrand für gerade Seiten beträgt 9 Leerzeichen vor dem linken Rand. Wenn Sie vorhaben, ein Buch zu schreiben, sollte der Heftrand für die geraden Seiten (dies sind normalerweise die linken Buchseiten) kleiner sein, so dass der Text nach links verschoben wird.

Leerzeilen am Seitenanfang

.TM - Leerzeilen über der Titelzeile

```
.TM zahl
```

Diese Anweisung definiert die Anzahl von Leerzeilen über der Titelzeile. Der Anfangswert ist .TM 0. Die Variable \$TM enthält den aktuellen Wert (siehe [Veränderbare Systemvariablen](#)).

Beispiel: Um vor der Titelzeile 2 Leerzeilen auszugeben, geben Sie an:

```
.TM 2
```

.HM - Leerzeilen zwischen Titelzeile und Text

```
.HM zahl
```

Diese Anweisung definiert die Anzahl von Leerzeilen zwischen Titelzeile und Kopfzeile bzw. zwischen Titelzeile und Text (wenn keine Kopfzeile definiert ist). Der Anfangswert ist .HM 2. Die Variable \$HM enthält den aktuellen Wert (siehe [Veränderbare Systemvariablen](#)).

Beispiel: Um zwischen der Titelzeile und der Kopfzeile 3 Leerzeilen zu definieren, geben Sie an:

```
.HM 3
```



Anmerkung: Die Anfangswerte für die Titelzeile (.HS 1) und den Abstand Titelzeile - Text (.HM 2) ergeben 3 Zeilen. Wenn also weder Titelzeile noch Kopfzeile definiert werden, erscheinen am Anfang des formatierten Textes immer 3 Leerzeilen.

Leerzeilen am Seitenende

.BM - Leerzeilen nach der Fußzeile

```
.BM zahl
```

Diese Anweisung definiert, wie viele Leerzeilen nach der Fußzeile folgen sollen. Die Variable \$BM enthält den aktuellen Wert für den unteren Rand (siehe [Veränderbare Systemvariablen](#)).

Diese Anweisung ist in jedem Fall redundant, da auf die Fußzeile immer ein Seitenvorschub folgt. Wenn Sie trotzdem am unteren Seitenrand 4 Leerzeilen definieren möchten, geben Sie an:

```
.BM 4
```

.FM - Leerzeilen zwischen Text und Fußzeile

```
.FM zahl
```

Diese Anweisung definiert, wie viele Leerzeilen zwischen dem Text (bzw. der letzten Fußnote, falls vorhanden) und der Fußzeile stehen sollen. Der Anfangswert ist .FM 2. Die Variable \$FM enthält den aktuellen Wert (siehe [Veränderbare Systemvariablen](#)).

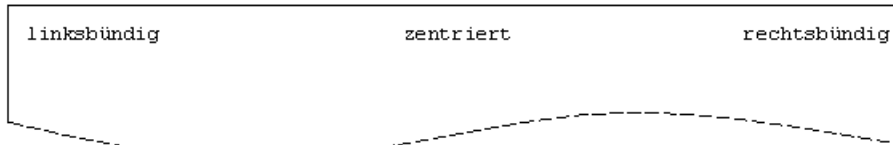
Beispiel: Wenn zwischen Text (oder Fußnoten) und Fußzeile 3 Leerzeilen ausgegeben werden sollen, geben Sie an:

```
.FM 3
```

Titelzeile und Fußzeile

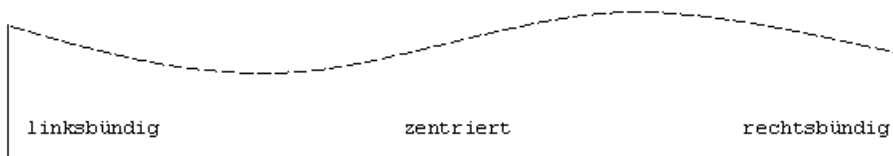
Die *Titelzeile* erscheint ganz oben auf jeder Seite. Sie kann aus mehreren Zeilen bestehen. Als Anfangswert ist eine Zeile für die Titelzeile reserviert (.HS 1; siehe [.HS - Zeilenreservierung für mehrzeilige Titelzeile](#)).

Der Text der Titelzeile kann linksbündig, zentriert oder rechtsbündig ausgedruckt werden:



Die *Fußzeile* erscheint unten auf jeder Seite. Sie kann aus mehreren Zeilen bestehen. Als Anfangswert ist eine Zeile für die Fußzeile reserviert (.FS 1; siehe [.FS - Zeilenreservierung für mehrzeilige Fußzeile](#)).

Der Text der Fußzeile kann linksbündig, zentriert oder rechtsbündig ausgegeben werden:



Der rechte und der linke Rand (mit den Anweisungen .RM und .LM definiert) gilt *nicht* für die Titelzeile und die Fußzeile. Deren Länge wird mit der Anweisung .LL bestimmt (siehe [.LL - Zeilenlänge](#)).

Sie müssen darauf achten, dass der Text für die Titel- und Fußzeile nicht zu lang wird; er wird sonst automatisch abgeschnitten, um ein Überlappen der drei Textbereiche (links, zentriert, rechts) zu vermeiden.

Für die Titelzeile und die Fußzeile gelten folgende Einschränkungen:

- Der Text darf keinen Schrägstrich (/) enthalten, da dieser als Trennzeichen für die drei Textbereiche dient.
- Der Parameter für den ersten Textbereich darf nicht mit einer Zahl beginnen, da Zahlen für mehrzeilige Titel- und Fußzeilen benutzt werden.
- Das Semikolon ist der Anfangswert für das Anweisungstrennzeichen. Wenn das Semikolon in der Titel- oder Fußzeile vorkommen soll, müssen Sie es deshalb doppelt eingeben (;).

- Ein Parameter kann höchstens 79 Zeichen haben. Dies gilt auch, wenn der Parameter Variablen enthält, d.h. nach dem Ersetzen der Variablen darf der Parameter nicht länger als 79 Zeichen sein.

Um die Seitenzahl auszugeben, müssen Sie das Nummernzeichen (#) im Parameter für den entsprechenden Textbereich eingeben (siehe [Seitennummerierung](#)).

Siehe auch: [Zeilenabstand und Schreibdichte am Seitenanfang und Seitenende](#).

Sie können eine Titelzeile und eine Fußzeile definieren, die für alle Seiten gelten, oder aber verschiedene Titel- und Fußzeilen für gerade und ungerade Seiten. Letzteres ist sinnvoll für ein Dokument, das als Buch gebunden werden soll.

Die im Folgenden beschriebenen Anweisungen für die Titel- und Fußzeilen haben das gleiche Parameterformat; es wird deshalb nur für die Anweisungen .TT (Titelzeile für alle Seiten) und .BT (Fußzeile für alle Seiten) erklärt.

Titelzeile

Wenn die folgenden Anweisungen angegeben werden, *bevor* der Text beginnt, erscheint die Titelzeile auf der ersten Seite des Ausdrucks. Wenn sie *nach* dem Text angegeben werden, erscheint die Titelzeile nicht auf der ersten Seite, sondern erst auf allen folgenden Seiten. (Dies gilt nicht für Fußzeilen).

.TT - Titelzeile für alle Seiten

```
.TT text/text/text
.TT
.TT zahl text/text/text
.TT zahl
```

Diese Anweisung definiert den Text, der auf allen Seiten (geraden und ungeraden) in der Titelzeile erscheinen soll.

Geben Sie in den Parametern der Anweisung .TT den für die Titelzeile gewünschten Text an. Um z.B. den Buchtitel links, die Seitenzahl in der Mitte und den Kapitelnamen rechts auszugeben, müssen Sie die folgende Anweisung eingeben:

```
.TT Con-form Benutzerhandbuch/#/Layout eines Dokuments
```

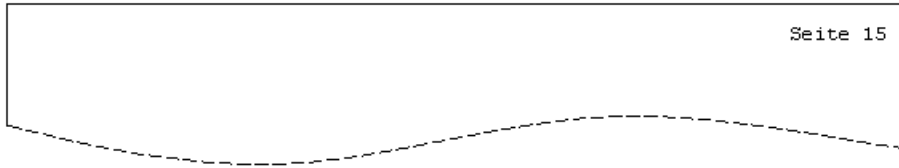
Mit dieser Anweisung wird folgende Titelzeile ausgegeben:

Con-form Benutzerhandbuch	1	Layout eines Dokuments
---------------------------	---	------------------------

Die drei Textbereiche müssen durch Schrägstriche getrennt werden. Es darf kein Schrägstrich weggelassen werden, sonst wird der Text nicht wie gewünscht formatiert. Dies ist vor allem dann wichtig, wenn Sie nur einen der drei Textbereiche angeben, wie z.B.:

```
.TT //Seite #
```

Wenn die aktuelle Seitenzahl 15 ist, sieht die Ausgabe folgendermaßen aus:



Mehrzeilige Titelzeile

Wenn der Titel aus mehreren Zeilen bestehen soll, müssen Sie zuerst die erforderlichen Zeilen reservieren (siehe [.HS - Zeilenreservierung für mehrzeilige Titelzeile](#)).

Für eine mehrzeilige Titelzeile muss vor jeder Textzeile eine Zeilennummer zwischen 1 und 9 eingegeben werden (ohne nachfolgendes Leerzeichen). Beispiel:

```
.TT 1//Dies ist die erste Zeile des Titels  
.TT 2//Dies ist die zweite Zeile des Titels
```

Sie können die Textzeilen in beliebiger Reihenfolge eingeben. Die Titelzeile kann aus höchstens 9 Textzeilen bestehen.

Titelzeile löschen

Wenn die Titelzeile auf den folgenden Seiten nicht mehr erscheinen soll, geben Sie die Anweisung ohne Parameter ein:

```
.TT
```

Um eine bestimmte Textzeile des mehrzeiligen Titels zu löschen, geben Sie die Anweisung mit der entsprechenden Zeilennummer ein, z.B.:

```
.TT 2
```

.ET - Titelzeile für gerade Seiten

```
.ET text/text/text  
.ET  
.ET zahl text/text/text  
.ET zahl
```

Diese Anweisung definiert den Text, der als Titelzeile auf allen geraden Seiten erscheinen soll. Sie wird genauso angewendet, wie die Anweisung **.TT**.

Die Titelzeile für gerade Seiten löschen

Um die Titelzeile für alle nachfolgenden geraden Seiten zu löschen, geben Sie die Anweisung ohne Parameter ein:

```
.ET
```

Um eine bestimmte Textzeile des mehrzeiligen Titels zu löschen, geben Sie die Anweisung mit der entsprechenden Zeilennummer ein, z.B.:

```
.ET 2
```

.OT - Titelzeile für ungerade Seiten

```
.OT text/text/text  
.OT  
.OT zahl text/text/text  
.OT zahl
```

Diese Anweisung definiert den Text, der als Titelzeile auf allen ungeraden Seiten erscheinen soll. Sie wird genauso angewendet, wie die Anweisung **.TT**.

Die Titelzeile für ungerade Seiten löschen

Um die Titelzeile für alle nachfolgenden ungeraden Seiten zu löschen, geben Sie die Anweisung ohne Parameter ein:

```
.OT
```

Um eine bestimmte Textzeile des mehrzeiligen Titels zu löschen, geben Sie die Anweisung mit der entsprechenden Zeilennummer ein, z.B.:

.OT 2

Fußzeile

.BT - Fußzeile für alle Seiten

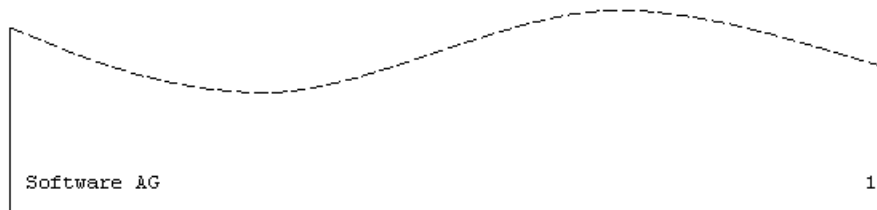
```
.BT text/text/text
.BT
.BT zahl text/text/text
.BT zahl
```

Diese Anweisung definiert den Text, der auf allen Seiten als Fußzeile ausgegeben werden soll.

Geben Sie in den Parametern der Anweisung `.BT` den für die Fußzeile gewünschten Text an. Um z.B. den Firmennamen links und die Seitenzahl rechts in der Fußzeile zu drucken, geben Sie Folgendes an:

```
.BT Software AG//#
```

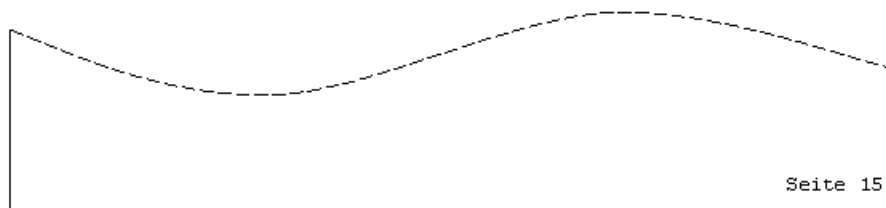
Die Fußzeile wird folgendermaßen ausgegeben:



Wie die Titelzeile hat auch die Fußzeile drei Textbereiche, die mit Schrägstrichen voneinander getrennt werden. Es darf kein Schrägstrich weggelassen werden, da der Text sonst nicht entsprechend formatiert wird. Dies ist besonders wichtig, wenn Sie nur einen Textbereich angeben wie im folgenden Beispiel:

```
.BT //Seite #
```

Wenn die aktuelle Seitenzahl 15 ist, wird mit dieser Anweisung die folgende Fußzeile ausgegeben:



Mehrzeilige Fußzeile

Wenn die Fußzeile mehrere Zeilen umfassen soll, müssen Sie zuerst die benötigten Zeilen reservieren (siehe [.FS - Zeilenreservierung für mehrzeilige Fußzeile](#)).

Um eine mehrzeilige Fußzeile zu definieren, müssen Sie eine Zeilennummer zwischen 1 und 9 vor dem Text angeben (ohne nachfolgendes Leerzeichen). Beispiel:

```
.BT 1//Dies ist die erste Zeile der Fußzeile  
.BT 2//Dies ist die zweite Zeile der Fußzeile
```

Die Textzeilen können in beliebiger Reihenfolge definiert werden. Es sind höchstens 9 Zeilen möglich.

Die Fußzeile löschen

Um die Fußzeile zu löschen, so dass sie auf den folgenden Seiten nicht mehr erscheint, geben Sie die Anweisung ohne Parameter ein:

```
.BT
```

Um eine bestimmte Zeile der mehrzeiligen Fußzeile zu löschen, geben Sie die Anweisung mit der entsprechenden Zeilennummer an, z.B.:

```
.BT 2
```

.EB - Fußzeile für gerade Seiten

```
.EB text/text/text  
.EB  
.EB zahl text/text/text  
.EB zahl
```

Diese Anweisung definiert den Text, der als Fußzeile auf allen geraden Seiten erscheinen soll.

Siehe die Beschreibung der Anweisung [.BT](#).

Die Fußzeile für gerade Seiten löschen

Um die Fußzeile zu löschen, so dass sie auf den folgenden geraden Seiten nicht mehr erscheint, geben Sie die Anweisung ohne Parameter ein:

```
.EB
```

Um eine bestimmte Zeile der mehrzeiligen Fußzeile zu löschen, geben Sie die Anweisung mit der entsprechenden Zeilennummer an, z.B.:

```
.EB 2
```

.OB - Fußzeile für ungerade Seiten

```
.OB text/text/text  
.OB  
.OB zahl text/text/text  
.OB zahl
```

Diese Anweisung definiert den Text, der als Fußzeile auf allen ungeraden Seiten erscheinen soll.

Siehe die Beschreibung der Anweisung [.BT](#).

Die Fußzeile für ungerade Seiten löschen

Um die Fußzeile zu löschen, so dass sie auf den folgenden ungeraden Seiten nicht mehr erscheint, geben Sie die Anweisung ohne Parameter ein:

.0B

Um eine bestimmte Zeile der mehrzeiligen Fußzeile zu löschen, geben Sie die Anweisung mit der entsprechenden Zeilennummer an, z.B.:

.0B 2

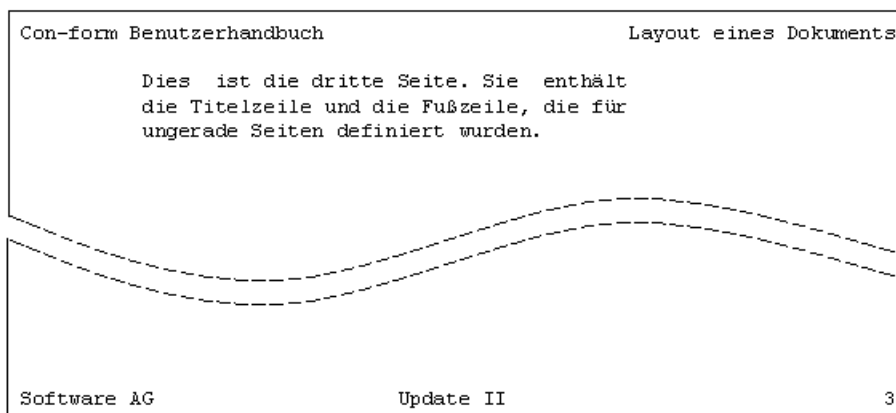
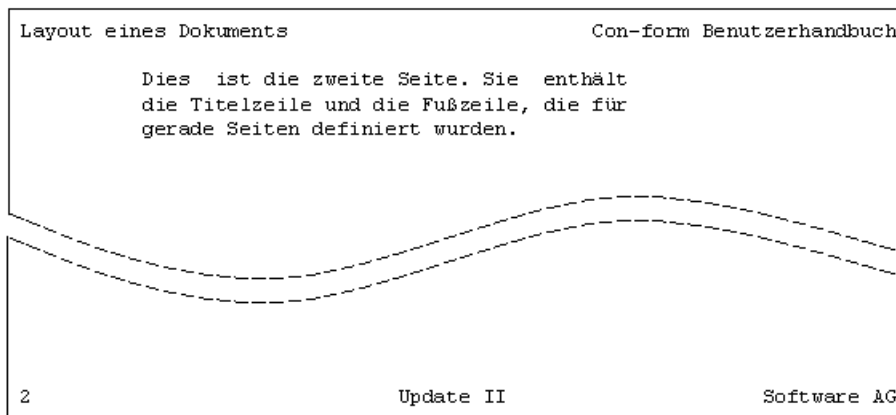
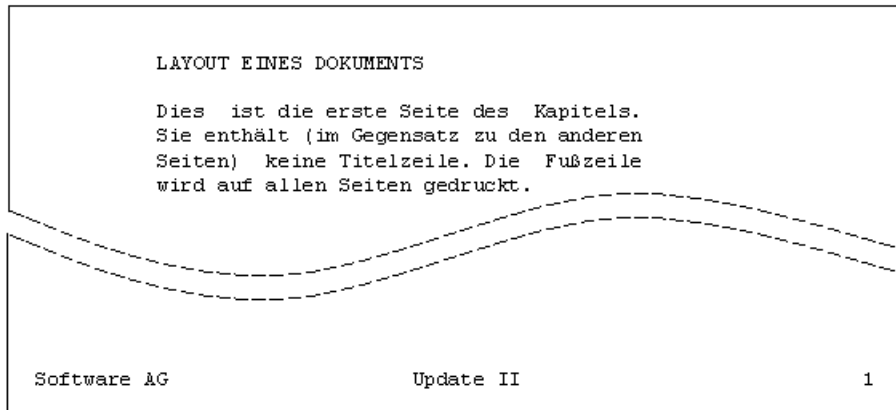
Beispiel

Dieses Beispiel zeigt die verschiedenen Anweisungen zum Definieren von Titelzeilen und Fußzeilen auf geraden und ungeraden Seiten. Beachten Sie, dass die erste Seite keine Titelzeile enthält.

Rohtext

```
.LM 10;.RM 50
LAYOUT EINES DOKUMENTS
.** Die Titelzeile wird NACH dem Kapitelnamen definiert.
.** Die erste Seite des Kapitels unterscheidet sich von den anderen Seiten.
.** Die Titelzeile wird auf allen folgenden Seiten gedruckt.
.** Diese Beschreibung gilt NICHT für die Fußzeile; diese wird auf allen
.** Seiten ausgedruckt.
.OT Con-form Benutzerhandbuch//Layout eines Dokuments
.ET Layout eines Dokuments//Con-form Benutzerhandbuch
.OB Software AG/Update II/#
.EB #/Update II/Software AG
.SL 1
.FI ON;.JU ON
Dies ist die erste Seite des Kapitels. Sie enthält (im Gegensatz zu den
anderen Seiten) keine Titelzeile. Die Fußzeile wird auf allen Seiten
gedruckt.
.NP
Dies ist die zweite Seite. Sie enthält die Titelzeile und die Fußzeile, die
für gerade Seiten definiert wurden.
.NP
Dies ist die dritte Seite. Sie enthält die Titelzeile und die Fußzeile, die
für ungerade Seiten definiert wurden.
```

Formatierte Ausgabe



Mehrzeilige Titel- und Fußzeilen

Wenn die Titelzeile oder die Fußzeile aus mehreren Zeilen bestehen soll, müssen Sie zuerst mit der Anweisung `.HS` bzw. `.FS` die benötigten Zeilen reservieren.

Wie der Text für mehrzeilige Titel- und Fußzeilen definiert wird, ist im Abschnitt [Titelzeile und Fußzeile](#) beschrieben.

.HS - Zeilenreservierung für mehrzeilige Titelzeile

```
.HS zahl
```

Diese Anweisung definiert, wie viele Zeilen für die Titelzeile reserviert werden sollen. Der Anfangswert ist `.HS 1` (d.h. wenn die Titelzeile aus nur einer Zeile bestehen soll, müssen Sie diese Anweisung nicht eingeben). Die Variable `$HS` enthält den aktuellen Wert (siehe [Veränderbare Systemvariablen](#)).

Wenn Sie z.B. 3 Zeilen reservieren möchten, geben Sie an:

```
.HS 3
```

Wenn Sie als Parameter eine Zahl angeben, die größer ist als die Anzahl der für die Titelzeile definierten Textzeilen, werden zwischen dem Titel und dem Rumpf des Dokuments entsprechend viele Leerzeilen ausgegeben.

Die Zeilenreservierung für die Titelzeile beeinflusst die Länge des Dokumentrumpfes: je länger die Titelzeile, desto kürzer der Rumpf des Dokuments.

.FS - Zeilenreservierung für mehrzeilige Fußzeile

```
.FS zahl
```

Diese Anweisung definiert, wie viele Zeilen für die Fußzeile reserviert werden sollen. Der Anfangswert ist `.FS 1` (d.h. wenn die Fußzeile aus nur einer Zeile bestehen soll, müssen Sie diese Anweisung nicht eingeben). Die Variable `$FS` enthält den aktuellen Wert (siehe [Veränderbare Systemvariablen](#)).

Wenn Sie z.B. 2 Zeilen reservieren möchten, geben Sie an:

.FS 2

Wenn Sie als Parameter eine Zahl angeben, die größer ist als die Anzahl der für die Fußzeile definierten Textzeilen, werden zwischen dem Rumpf des Dokuments und der Fußzeile entsprechend viele Leerzeilen ausgegeben.

Die Zeilenreservierung für die Fußzeile beeinflusst die Länge des Dokumentrumpfes nicht.

Beispiel

Dieses Beispiel zeigt, wie mehrzeilige Titel- und Fußzeilen auf geraden und ungeraden Seiten angelegt werden. Es enthält auch die Anweisung **.LL**, die die Länge der Titel- und Fußzeile definiert, sowie die Anweisung **.NP**, die einen Seitenvorschub bewirkt.

Rohtext

```
.** Für die Titelzeile und die Fußzeile werden je 2 Zeilen reserviert
.HS 2;.FS 2
.** Die Zeilenlänge für die Titel- und Fußzeile ist:
.LL 60
.OT 1//Dies ist die Titelzeile für ungerade Seiten
.OT 2//Diese Zeile erscheint nur auf ungeraden Seiten
.ET Dies ist die Titelzeile für gerade Seiten//
.OB 1Software AG//Update II
.OB 2//Seite #
.EB 1Update II//Software AG
.EB 2Seite #//
.SL 1
.LM 10;.RM 50
.FI ON;.JU ON
.** Dieser Seite wird die Seitenzahl 3 zugewiesen:
.NP 3
Dies ist eine ungerade Seite. Sie hat eine rechtsbündige Titelzeile, die
aus 2 Zeilen besteht. Diese beiden Zeilen für die mehrzeilige Titelzeile
wurden mit der Anweisung .HS reserviert.
.SL 1
Die Fußzeile besteht aus 2 Zeilen und zeigt die aktuelle Seitenzahl rechts
an.
.NP
Dies ist eine gerade Seite. Sie hat eine linksbündige Titelzeile. Diese
Titelzeile besteht (im Gegensatz zur Titelzeile für ungerade Seiten) aus
einer Zeile. Da aber .HS 2 definiert wurde, wird unter dieser Titelzeile
eine zusätzliche Leerzeile ausgegeben.
.SL 1
Die Fußzeile dieser Seite besteht ebenfalls aus 2 Zeilen. Sie zeigt die
aktuelle Seitenzahl links an.
```

Formatierte Ausgabe

Dies ist die Titelzeile für ungerade Seiten
Diese Zeile erscheint nur auf ungeraden Seiten

Dies ist eine ungerade Seite. Sie hat eine rechtsbündige Titelzeile, die aus 2 Zeilen besteht. Diese beiden Zeilen für die mehrzeilige Titelzeile wurden mit der Anweisung .HS reserviert.

Die Fußzeile besteht aus 2 Zeilen und zeigt die aktuelle Seitenzahl rechts an.

Software AG Update II
Seite 3

Dies ist die Titelzeile für gerade Seiten

Dies ist eine gerade Seite. Sie hat eine linksbündige Titelzeile. Diese Titelzeile besteht (im Gegensatz zur Titelzeile für ungerade Seiten) aus einer Zeile. Da aber .HS 2 definiert wurde, wird unter dieser Titelzeile eine zusätzliche Leerzeile ausgegeben.

Die Fußzeile dieser Seite besteht ebenfalls aus 2 Zeilen. Sie zeigt die aktuelle Seitenzahl links an.

Update II Software AG
Seite 4

Länge der Titel- und Fußzeilen

Die Titelzeile und die Fußzeile beginnen direkt nach dem Heftrand. Ihre Länge wird mit der Anweisung `.LL` definiert.

Linker und rechter Rand (definiert mit den Anweisungen `.LM` und `.RM`) gelten nicht für die Titelzeile und die Fußzeile.

.LL - Zeilenlänge

```
.LL zahl
```

Diese Anweisung definiert die Zeilenlänge der Titelzeile und der Fußzeile. Der Anfangswert ist `.LL 72`. Der zulässige Höchstwert ist 190. Die Variable `$LL` enthält den aktuellen Wert (siehe [Veränderbare Systemvariablen](#)).

Um die Zeilenlänge zu definieren, geben Sie die Anzahl der Zeichen als Parameter der Anweisung an, z.B.:

```
.LL 60
```

Wenn die Ausgabezeile länger ist als die mit dieser Anweisung definierte Länge, gibt Con-form eine Fehlermeldung aus.

Beispiel

Dieses Beispiel zeigt, wie die Länge der Titel- und Fußzeilen definiert wird. Es enthält außerdem die Anweisung **.PL**, mit der die Seitenlänge festgelegt wird. Diese Anweisungen sind wichtig, wenn Sie ein Buch in einem kleineren Format produzieren möchten. Die zweite Seite wird automatisch auf einem neuen Blatt Papier gedruckt. Wenn das Dokument im Buchformat gebunden wird, müssen Sie das Papier dem Seitenformat entsprechend schneiden.

Rohtext

```
.LL 55;.PL 30
.HS 2;.FS 2
.OT 1Con-form//Layout eines Dokuments
.OT 2Benutzerhandbuch//
.ET 1Layout eines Dokuments//Con-form
.ET 2Benutzerhandbuch
.OB 1Software AG//Update II
.OB 2//Seite #
.EB 1Update II//Software AG
.EB 2Seite #//
.SL 1
.LM 5;.RM 50
.FI ON;.JU ON
```

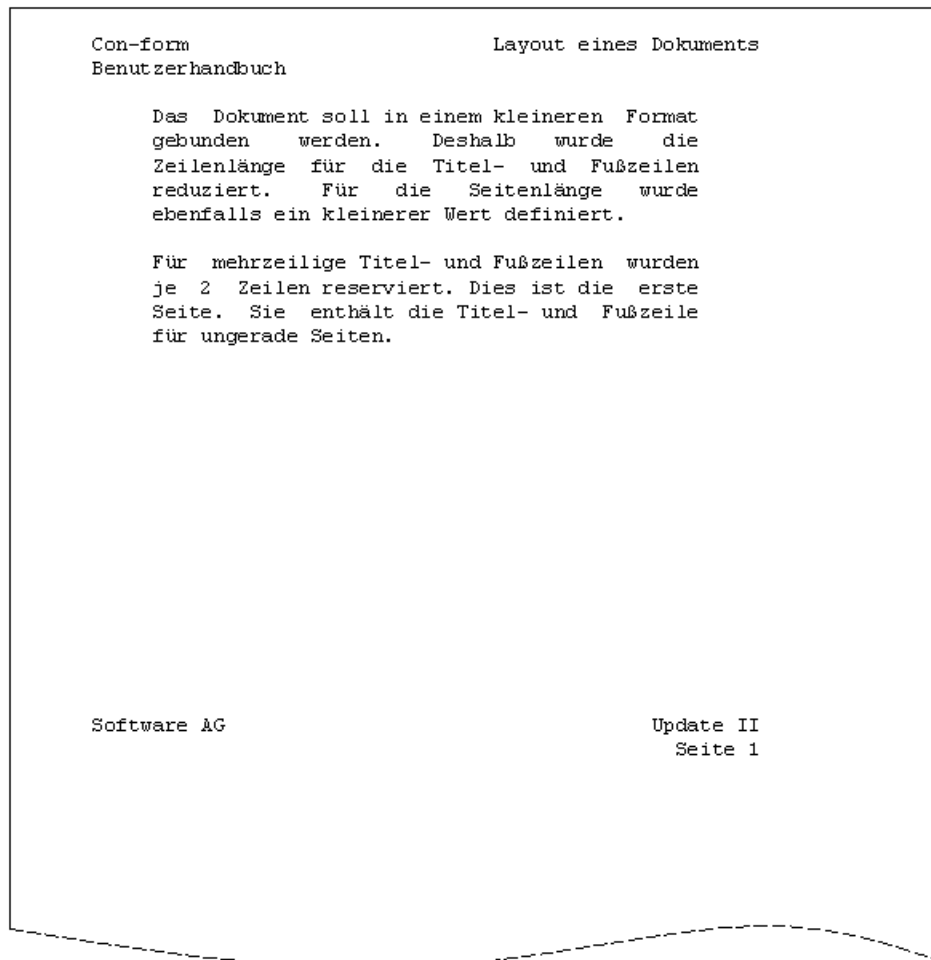
Das Dokument soll in einem kleineren Format gebunden werden. Deshalb wurde die Zeilenlänge für die Titel- und Fußzeilen reduziert. Für die Seitenlänge wurde ebenfalls ein kleinerer Wert definiert.

```
.SL 1
```

Für mehrzeilige Titel- und Fußzeilen wurden je 2 Zeilen reserviert. Dies ist die erste Seite. Sie enthält die Titel- und Fußzeile für ungerade Seiten.

```
.NP
.** Der Text für die folgenden Seiten würde im Anschluß an diesen Kommentar
.** eingegeben.
```

Formatierte Ausgabe



Seitennummerierung

Für die Seitenzahl ist anfänglich das Nummernzeichen (#) definiert. Sie können es in der Titelzeile und der Fußzeile sowie mit der Anweisung **.SV** benutzen. In der Fußzeile repräsentiert das Nummernzeichen entweder die aktuelle oder die nächste Seitenzahl (siehe **.OP PAG - Aktuelle oder nächste Seitenzahl**). In der Titelzeile repräsentiert es die aktuelle Seitenzahl.

Die Variable \$PN enthält die aktuelle Seitenzahl (siehe **Veränderbare Systemvariablen**).

.OP PNS - Zeichen der Seitennummerierung

```
.OP PNS=zeichen
```

Wenn Sie als Zeichen für die Seitennummerierung nicht das Nummernzeichen benutzen möchten, können Sie es mit dieser Anweisung durch ein anderes Zeichen ersetzen. Beispiel: Um das Ausrufezeichen (!) als Zeichen für die Seitennummerierung zu definieren, geben Sie an:

```
.OP PNS=!
```

Zum Ausgeben der Seitenzahl in der Titel- oder Fußzeile müssen Sie nun das Ausrufezeichen anstelle des Nummernzeichens benutzen.

Wenn Sie das Zeichen für die Seitennummerierung im Dokumentrumpf benutzen, wird es als normaler Text interpretiert (d.h. in der formatierten Ausgabe erscheint das Zeichen und nicht die Seitenzahl).

.OP PAG - Aktuelle oder nächste Seitenzahl

```
.OP PAG=EQU  
.OP PAG=DIF
```

Diese Anweisung gilt nur für die Seitenzahl in der Fußzeile. Der Anfangswert dieser Option ist EQU.

Die folgende Anweisung definiert, dass die *aktuelle* Seitenzahl in der Fußzeile erscheinen soll:

```
.OP PAG=EQU
```

Die folgende Anweisung definiert, dass die *nächste* Seitenzahl in der Fußzeile erscheinen soll (bis auf die letzte Seite, die keine Fußzeile enthält):

```
.OP PAG=DIF
```

Beispiel

Dieses Beispiel zeigt, wie Sie mit der Anweisung `.OP PAG` die Seitenzahl der nächsten Seite in der Fußzeile ausgeben. Dies ist sinnvoll, um darauf hinzuweisen, dass nachfolgende Seiten vorhanden sind.

Rohtext

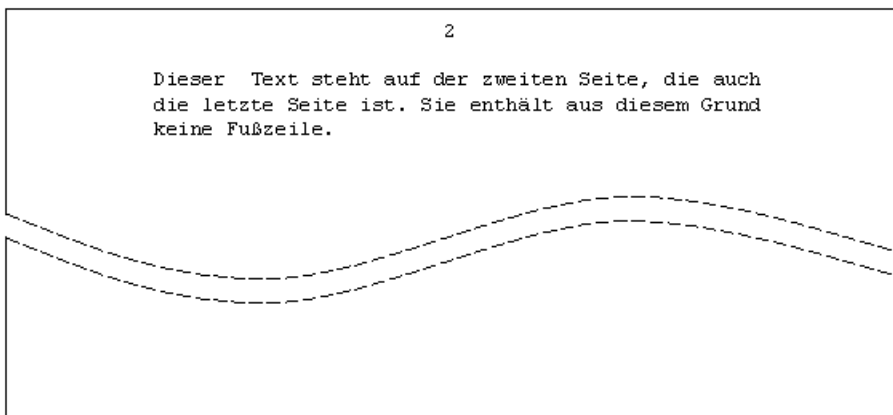
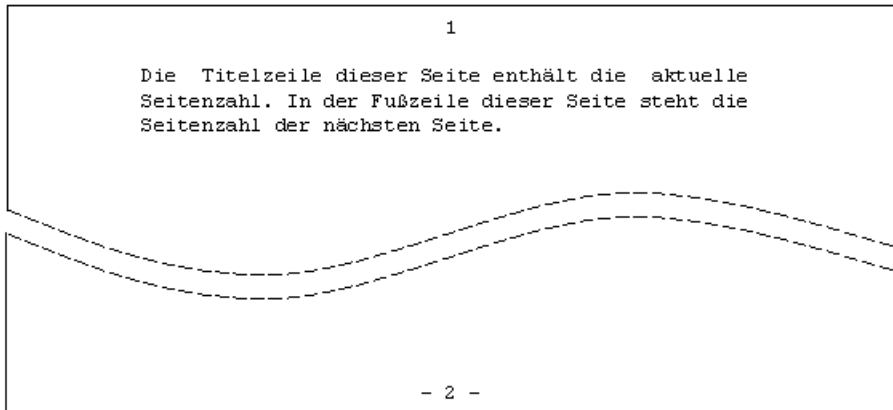
```
.OP PAG=DIF
.TT /#/
.BT /- # - /
.LM 10;.RM 60
.FI ON;.JU ON
```

Die Titelzeile dieser Seite enthält die aktuelle Seitenzahl. In der Fußzeile dieser Seite steht die Seitenzahl der nächsten Seite.

```
.NP
```

Dieser Text steht auf der zweiten Seite, die auch die letzte Seite ist. Sie enthält aus diesem Grund keine Fußzeile.

Formatierte Ausgabe



.PM - Form der Seitennummerierung

```
.PM A  
.PM R  
.PM r
```

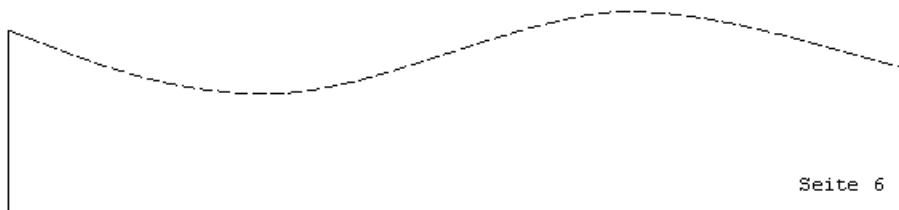
Die Seitenzahl kann entweder in arabischen oder römischen Zahlen ausgegeben werden. Der Anfangswert ist arabisch.

Arabische Seitenzahlen

Um arabische Seitenzahlen zu definieren, geben Sie an:

```
.PM A
```

Die Seitenzahl wird mit einer arabischen Zahl ausgegeben:



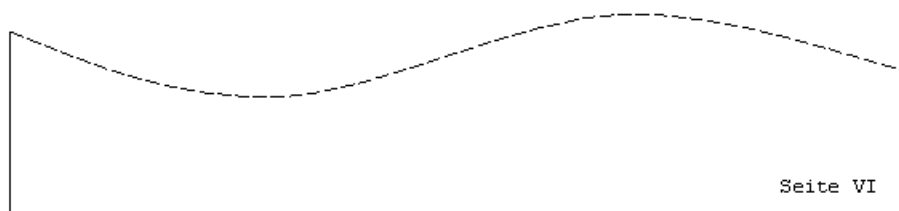
Römische Seitenzahlen

Die römischen Seitenzahlen können in Groß- oder Kleinschreibung ausgegeben werden. Um römische Seitenzahlen zu definieren, die in Großbuchstaben erscheinen sollen, geben Sie an:

```
.PM R
```

Achten Sie darauf, den Parameter als Großbuchstaben anzugeben.

Die Seitenzahl wird mit einer römischen Zahl in Großbuchstaben ausgegeben:

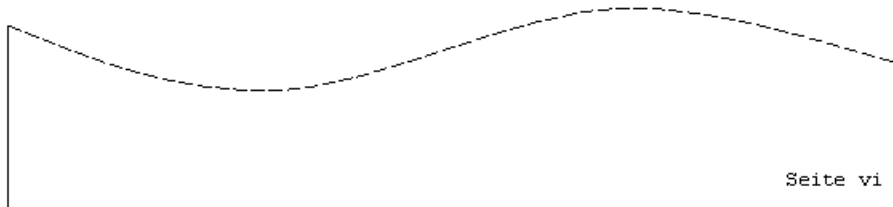


Um römische Seitenzahlen zu definieren, die in Kleinbuchstaben erscheinen sollen, geben Sie an:

```
.PM r
```

Achten Sie darauf, den Parameter als Kleinbuchstaben anzugeben.

Die Seitenzahl wird als römische Zahl in Kleinbuchstaben ausgegeben:



.OP ROM - Groß-/Kleinbuchstaben bei römischen Seitenzahlen

```
.OP ROM=UPPER  
.OP ROM=LOWER
```

Der Anfangswert dieser Option ist UPPER (d.h. römische Seitenzahlen werden in Großbuchstaben ausgegeben).

Anstelle der Anweisung .PM R können Sie auch die folgende Anweisung angeben, um Seitenzahlen in Großbuchstaben zu definieren:

```
.OP ROM=UPPER
```

Anstelle der Anweisung .PM r können Sie auch die folgende Anweisung angeben, um Seitenzahlen in Kleinbuchstaben zu definieren:

```
.OP ROM=LOWER
```

Kopfzeilen

Die Kopfzeile erscheint oben auf jeder Seite zwischen der Titelzeile und dem Dokumentrumpf. Sie können die Kopfzeile auch im Dokumentrumpf ausgeben.

Die Anzahl der Leerzeilen zwischen Titelzeile und Kopfzeile wird durch den Abstand zwischen Titelzeile und Text bestimmt (Anfangswert ist `.HM 2`, siehe [.HM - Leerzeilen zwischen Titelzeile und Text](#)).

Anders als die Titel- und Fußzeile wird die Kopfzeile beeinflusst von den Werten, die für den Rumpf des Dokuments gelten (wie z.B. linker und rechter Rand). Der Dokumentrumpf wird nicht von den Werten beeinflusst, die für die Kopfzeile definiert wurden.

Siehe auch: [Zeilenabstand und Schreibdichte in Kopfzeilen](#).

.HL - Kopfzeile

```
.HL anweisung  
.HL text
```

Die Anweisung `.HL` kann auf zwei Arten benutzt werden: entweder Sie geben als Parameter den Text an, der in der Kopfzeile erscheinen soll, oder Sie geben eine Anweisung an, die für die Kopfzeile gelten soll. Beides in einer `.HL`-Anweisung zu mischen, ist nicht möglich.

Um den Text für die Kopfzeile zu definieren, geben Sie an:

```
.HL Dies ist die Kopfzeile
```

Um eine Anweisung für die Kopfzeile zu definieren, z.B. die Kopfzeile zentrieren, geben Sie an:

```
.HL .CE ON
```

Wenn Sie eine neue Kopfzeile definieren möchten, müssen Sie zuerst die aktuelle Kopfzeile löschen (mit der Anweisung `.HC`) und dann die neue Kopfzeile mit der Anweisung `.HL` definieren.

.HC - Kopfzeile löschen

```
.HC
```

Diese Anweisung löscht die Kopfzeile, die mit der Anweisung .HL definiert wurde. Um die aktuelle Kopfzeile zu löschen, geben Sie an:

```
.HC
```

.PH - Kopfzeile im Text drucken

```
.PH
```

Diese Anweisung gibt die Kopfzeile, die mit der Anweisung `.HL` definiert wurde, an der aktuellen Position im Text aus. Die Anweisungen, die für die Kopfzeile definiert wurden, werden dabei berücksichtigt. Wenn die Kopfzeile z.B. mit der Anweisung `.HL .CE ON` zentriert wurde, wird sie auch im Dokumentrumpf zentriert.

Um die Kopfzeile an der aktuellen Textposition zu drucken, geben Sie an:

```
.PH
```

Sie können auch eine Kopfzeile an einer beliebigen Stelle innerhalb des Textes definieren und dann die Anweisung `.PH` angeben, um die Kopfzeile sofort auszugeben. Die Kopfzeile erscheint dann automatisch oben auf den nachfolgenden Seiten (sofern sie nicht mit der Anweisung `.HC` gelöscht wird).

Da die Kopfzeile automatisch oben auf jeder Seite erscheint, ist die Anweisung `.PH` nur nötig, wenn Sie die Kopfzeile im Dokumentrumpf drucken möchten.

Beispiel

Dieses Beispiel zeigt, wie eine Kopfzeile definiert und im Dokumentrumpf gedruckt wird. Es enthält außerdem die Anweisung `.CE`, mit der Text zentriert werden kann.

Rohtext

```
.LL 60
.TT Con-form Benutzerhandbuch//Seite #
.FI ON;.JU ON
.LM 0;.RM 60
.HL .CE ON
.HL Dies ist die Kopfzeile.
.HL Sie wird oben auf jeder Seite gedruckt.
.HL Diese Kopfzeile ist zentriert.
.HL .SL 1
.** Sie sollten mit der Anweisung .HL immer eine Leerzeile definieren,
.** damit die Kopfzeile und der eigentliche Text immer getrennt sind.
Diese Zeilen bilden den Rumpf des Dokuments.
Beachten Sie, dass zwischen der Titelzeile und der Kopfzeile automatisch
2 Leerzeilen eingefügt wurden. Dies wurde durch den Anfangswert für den
Abstand zwischen Titelzeile und Text (.HM 2) bewirkt.
.SL 1
Sie können die Kopfzeilen im Rumpf des Dokuments ausdrucken. Die Anweisung,
die für die Kopfzeilen definiert wurde, wird berücksichtigt.
.SL 1
.PH
.** Sie brauchen hier keine Leerzeile zu definieren, da schon durch die
.** Anweisung .HL eine Leerzeile für die Kopfzeilen definiert wurde.
Dieser Text folgt der Kopfzeile, die im Rumpf des Dokuments gedruckt wurde.
Er wird automatisch zwischen den definierten Rändern angeordnet.
```

Formatierte Ausgabe

Con-form Benutzerhandbuch

Seite 1

Dies ist die Kopfzeile.
Sie wird oben auf jeder Seite gedruckt.
Diese Kopfzeile ist zentriert.

Diese Zeilen bilden den Rumpf des Dokuments. Beachten Sie, dass zwischen der Titelzeile und der Kopfzeile automatisch 2 Leerzeilen eingefügt wurden. Dies wurde durch den Anfangswert für den Abstand zwischen Titelzeile und Text (.HM 2) bewirkt.

Sie können die Kopfzeilen im Rumpf des Dokuments ausdrucken. Die Anweisung, die für die Kopfzeilen definiert wurde, wird berücksichtigt.

Dies ist die Kopfzeile.
Sie wird oben auf jeder Seite gedruckt.
Diese Kopfzeile ist zentriert.

Dieser Text folgt der Kopfzeile, die im Rumpf des Dokuments gedruckt wurde. Er wird automatisch zwischen den definierten Rändern angeordnet.

Fußnoten

.FN - Fußnote

```
.FN text  
.FN ON  
.FN OFF
```

Sie können Text definieren, der als Fußnote zwischen dem Dokumentrumpf und der Fußzeile gedruckt werden soll. Über der ersten Fußnote druckt Con-form automatisch eine kurze Linie aus.

Die Anzahl der Leerzeilen zwischen den Fußnoten und der Fußzeile wird durch den Abstand zwischen Text und Fußzeile bestimmt (Anfangswert `.FM 2`; siehe [.FM - Leerzeilen zwischen Text und Fußzeile](#)).

Siehe auch: [Zeilenabstand und Schreibdichte in Fußnoten](#).

Die Anweisung `.FN` kann auf zwei Arten benutzt werden.

Einzeilige Fußnote

Um eine Fußnote zu definieren, deren Text aus einer Zeile besteht, geben Sie die Anweisung `.FN` mit dem gewünschten Text an, z.B.:

```
.FN Dies ist die Fußnote.
```

Wenn mehrere dieser Anweisungen eingegeben werden, wird pro Anweisung eine Zeile ausgegeben.

Die Fußnoten werden nicht beeinflusst von den Werten, die für den Rumpf des Dokuments definiert wurden (wie z.B. linker und rechter Rand). Die Fußnote beginnt immer in Spalte 0. Sie müssen darauf achten, dass der Text der Fußnote nicht so lang wird, dass er nicht mehr auf die Seite passt. Wie eine Fußnote als Textblock ausgegeben wird, ist im Folgenden beschrieben. In diesem Fall können Sie auch ein Layout für die Fußnote definieren.

Mehrzeilige Fußnote

Sie können die Fußnote als Textblock ausgeben. Dazu müssen Sie den Fußnotentext und alle für die Fußnote geltenden Anweisungen zwischen den beiden folgenden Anweisungen angeben:

```
.FN ON
```

```
.FN OFF
```

Wenn Sie diese Anweisungen verwenden, können Sie den linken und rechten Rand sowie Zeilenausgleich und Randausgleich für die Fußnote definieren.

Der Zeilenausgleich ist für die Fußnote anfänglich eingeschaltet. Wenn Sie den linken und rechten Rand nicht definieren, gelten die Anfangswerte (.LM 0 und .RM 72).

Wenn Sie den Randausgleich für die Fußnote nicht definieren, gilt die Einstellung, die für den Dokumentrumpf definiert wurde. Ist der Randausgleich für den Dokumentrumpf eingeschaltet, gilt er auch für die Fußnote.

Das für eine Fußnote definierte Layout gilt nur für diese eine Fußnote. Soll die nächste Fußnote genauso ausgegeben werden, müssen Sie das Layout nochmals definieren.

Mit einer **.IF-Anweisung** ist es nicht möglich, die Verarbeitung von .FN OFF bedingt zu unterdrücken.

Fußnoten nummerieren

Con-form nummeriert die Fußnoten nicht automatisch. Wenn vor einer Fußnote eine Zahl erscheinen soll, müssen Sie dies explizit definieren.

Manuelle Nummerierung

Wenn Sie die Fußnotennummern selbst angeben, hat dies den Nachteil, dass Sie alle Fußnoten neu durchnummerieren müssen, wenn Sie eine Fußnote nachträglich einfügen.

Im folgenden Beispiel wurde "/1/" in den Text des Dokuments eingefügt, damit klar hervorgeht, zu welchem Textabschnitt die Fußnote gehört.

```
Dies gehört zum Rumpf des Dokuments /1/  
.FN /1/ Dies ist die erste Fußnote.  
und dies ist die Fortsetzung des Dokumentrumpfes.
```

Der Text wird folgendermaßen formatiert (innerhalb der definierten Ränder):

```
Dies gehört zum Rumpf des Dokuments /1/ und dies ist die Fortsetzung  
des Dokumentrumpfes.
```

Die Fußnote wird folgendermaßen formatiert:

```
/1/ Dies ist die erste Fußnote.
```

Automatische Nummerierung

Sie können die Fußnoten von Con-form nummerieren lassen. Die Variable \$FN enthält die Nummer der aktuellen Fußnote (siehe [Veränderbare Systemvariablen](#)). Jedesmal wenn die Variable \$FN im Text verarbeitet wird, erhöht sich ihr Wert um eins.

Um die Fußnoten fortlaufend zu nummerieren, müssen Sie den Wert der Variablen \$FN mit der Anweisung **.SV** einer anderen Variablen zuweisen. Sie müssen die Anweisung **.SV** für jede Fußnote wiederholen. Beispiel:

```
.SV nummer=&$FN
Dies gehört zum Rumpf des Dokuments /&nummer/
.FN /&nummer/ Dies ist die erste Fußnote.
und dies ist die Fortsetzung des Dokumentrumpfes.
.SL 1
.SV nummer=&$FN
Vergessen Sie nicht, die Variable "nummer" zu aktualisieren
.FN /&nummer/ Dies ist die zweite Fußnote.
/&nummer/, sonst wird die Fußnotennummer nicht erhöht.
```

Der Text wird folgendermaßen formatiert (innerhalb der definierten Ränder):

```
Dies gehört zum Rumpf des Dokuments /1/ und dies ist die Fortsetzung
des Dokumentrumpfes.

Vergessen Sie nicht, die Variable "nummer" zu aktualisieren /2/, sonst
wird die Fußnotennummer nicht erhöht.
```

Die Fußnoten werden folgendermaßen formatiert:

```
/1/ Dies ist die erste Fußnote.
/2/ Dies ist die zweite Fußnote.
```

Beispiel

Dieses Beispiel zeigt verschiedene Möglichkeiten zum Definieren von Fußnoten. Es zeigt, wie Fußnoten automatisch nummeriert werden und wie das Layout für eine Fußnote angegeben wird, die als Textblock definiert wurde.

Rohtext

```
.LM 5;.RM 60
.FI ON;.JU ON
Der folgende Text enthält Fußnoten. Die Fußnote wird immer auf der Seite
gedruckt, die auch die Referenz auf die Fußnote enthält.
.** &$FN ist eine spezielle Variable, die jedesmal, wenn sie benutzt wird,
.** hochgesetzt wird. Sie dient zum fortlaufenden Nummerieren der Fußnoten
.** innerhalb eines Dokuments.
Um eine Referenz auf die Fußnote innerhalb des laufenden Textes einzufügen,
müssen Sie eine Variable definieren, in der der aktuelle Wert der Fußnote
gespeichert wird. Damit können Sie den korrekten Wert in den Abschnitt
der Seite bringen, der auf die untenstehende Fußnote zeigt.
.SV wert=&$FN
.** Sie müssen die obige Anweisung für jede Fußnote wiederholen, die Sie
.** in Ihren Text einfügen.
Die erste Fußnote gilt für diesen Satz
.FN /&wert/Dies ist die erste Fußnote.
.** Die folgende Zeile enthält den Rest des Satzes, nämlich den Wert der
.** Fußnote und den Punkt.
/&wert/.
.IL 1
.** Der folgende Abschnitt enthält eine weitere Fußnote, deshalb muss
.** die Variable "wert" aktualisiert werden:
.SV wert=&$FN
Wenn Sie weitere Fußnoten erzeugen möchten, dürfen Sie nicht vergessen,
die Variable "wert" /&wert/ zu aktualisieren.
.FN /&wert/Dies ist die zweite Fußnote.
Es ist auch möglich, die Variable "wert" in der Mitte
.SV wert=&$FN
.FN /&wert/Dies ist die dritte Fußnote.
des Satzes zu aktualisieren /&wert/.
.NP
.SV wert=&$FN
Dies ist ein Beispiel eines Textblocks, der als Fußnote definiert wurde
/&wert/.
.FN ON
.** Sie können den linken und den rechten Rand für die Fußnote definieren.
.** Außerdem können Sie Zeilenausgleich und Randausgleich angeben. Diese
.** Einstellungen gelten nur für diese Fußnote.
.LM 10;.RM 50
.FI ON;.JU ON
.OF +4
/&wert/ Dies ist eine längere Fußnote. Der linke und der rechte Rand
```

wurden speziell für diese Fußnote definiert (andernfalls wären die Anfangswerte `.LM 0` und `.RM 72` benutzt worden). Außerdem wurden die Zeilen nach der ersten Zeile um 4 Spalten nach rechts verschoben.

Um diese Verschiebung wieder rückgängig zu machen, muss die Anweisung `.OF` noch einmal angegeben werden, dieses Mal aber ohne Parameter.

```
.OF
```

```
.FN OFF
```

Der Zeilenausgleich und der Randausgleich (`.FI` und `.JU`), die für die Fußnote definiert wurden, gelten nicht für den Rumpf des Dokuments.

Erste Seite der formatierten Ausgabe

Der folgende Text enthält Fußnoten. Die Fußnote wird immer auf der Seite gedruckt, die auch die Referenz auf die Fußnote enthält. Um eine Referenz auf die Fußnote innerhalb des laufenden Textes einzufügen, müssen Sie eine Variable definieren, in der der aktuelle Wert der Fußnote gespeichert wird. Damit können Sie den korrekten Wert in den Abschnitt der Seite bringen, der auf die untenstehende Fußnote zeigt. Die erste Fußnote gilt für diesen Satz /1/.

Wenn Sie weitere Fußnoten erzeugen möchten, dürfen Sie nicht vergessen, die Variable "wert" /2/ zu aktualisieren. Es ist auch möglich, die Variable "wert" in der Mitte des Satzes zu aktualisieren /3/.

/1/Dies ist die erste Fußnote.
/2/Dies ist die zweite Fußnote.
/3/Dies ist die dritte Fußnote.

Zweite Seite der formatierten Ausgabe

Dies ist ein Beispiel eines Textblocks, der als Fußnote definiert wurde /4/. Der Zeilenausgleich und der Randausgleich (.FI und .JU), die für die Fußnote definiert wurden, gelten nicht für den Rumpf des Dokuments.

_____/4/ Dies ist eine längere Fußnote. Der linke und der rechte Rand wurden speziell für diese Fußnote definiert (andernfalls wären die Anfangswerte .LM 0 und .RM 72 benutzt worden). Außerdem wurden die Zeilen nach der ersten Zeile um 4 Spalten nach rechts verschoben. Um diese Verschiebung wieder rückgängig zu machen, muss die Anweisung .OF noch einmal angegeben werden, dieses Mal aber ohne Parameter.

Seitenlänge

.PL - Seitenlänge

```
.PL zahl
```

Diese Anweisung bestimmt, wie viele Zeilen auf einer Seite gedruckt werden. Der Anfangswert ist .PL 57. Die Variable \$PL enthält die aktuelle Seitenlänge (siehe [Veränderbare Systemvariablen](#)).

Die Seitenlänge umfasst die Leerzeilen über der Titelzeile (.TM), die Titelzeile (d.h. die Anzahl Zeilen, die mit .HS dafür reserviert wurden), den Abstand zwischen Titelzeile und Text (.HM), den Text selbst sowie die Fußnoten.

Für die Seitenlänge *nicht* mitgezählt werden: Abstand zwischen Text und Fußzeile (.FM), unterer Rand (.BM) sowie die Zeilen, die für die Fußzeile reserviert wurden (.FS).

Um die Seitenlänge zu definieren, geben Sie die gewünschte Zeilenzahl an. Wenn Sie z.B. auf kleineres Papier drucken möchten und deshalb nur 40 Zeilen pro Seite gedruckt werden sollen, geben Sie an:

```
.PL 40
```

Die Definition der Seitenlänge kann nur am Anfang einer Seite berücksichtigt werden. Sie wird also nicht sofort ab der Position wirksam, an der die Anweisung .PL im Rohtext steht. Die neue Seitenlänge wird erst nach dem nächsten Seitenumbruch aktiviert. Wenn Sie die Seitenlänge für das gesamte Dokument definieren möchten, müssen Sie sie vor dem Dokumenttext angeben.

Zeilenabstand und Schreibdichte

.LS - Zeilenabstand

```
.LS S  
.LS D  
.LS zahl
```

Mit dieser Anweisung definieren Sie den Abstand der Zeilen in der formatierten Ausgabe.

Einfacher Zeilenabstand

Anfänglich ist der einfache Zeilenabstand wirksam. Dies entspricht der Anweisung:

```
.LS S
```

Doppelter Zeilenabstand

Um den doppelten Zeilenabstand zu aktivieren, geben Sie an:

```
.LS D
```

Zeilenanzahl pro Zoll

Diese Funktion wird in Verbindung mit einem Con-nect-Druckerprofil benutzt (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Druckerprofile*). Sie müssen Ihre Definitionen für den Zeilenabstand in das Druckerprofil aufnehmen, damit die Anweisung .LS die gewünschte Ausgabe liefert. Beispiel: Für die Anweisung .LS 060 müssen Sie Folgendes in Ihr Druckerprofil eingeben:

```
LI060
```

Sie können festlegen, wie viele Zeilen pro Zoll gedruckt werden sollen. Der Anfangswert ist 6 Zeilen pro Zoll, was der folgenden Anweisung entspricht:

```
.LS 060
```

Der Parameter *zahl* muss mit genau drei Stellen angegeben werden. Führende Nullen (wie im obigen Beispiel) müssen immer angegeben werden.

Einige Beispiele zum Definieren der Zeilenanzahl pro Zoll:

```
.LS 030
Diese Anweisung definiert 3 Zeilen pro Zoll.
Dies wurde mit der Anweisung .LS 030 erreicht.
.SL 1
.LS 040
Diese Anweisung definiert 4 Zeilen pro Zoll.
Dies wurde mit der Anweisung .LS 040 erreicht.
.SL 1
.LS 060
Diese Anweisung definiert 6 Zeilen pro Zoll.
Dies wurde mit der Anweisung .LS 060 erreicht.
.SL 1
.LS 080
Diese Anweisung definiert 8 Zeilen pro Zoll.
Dies wurde mit der Anweisung .LS 080 erreicht.
.SL 1
.LS 120
Diese Anweisung definiert 12 Zeilen pro Zoll.
Dies wurde mit der Anweisung .LS 120 erreicht.
```

Beispiel

Dieses Beispiel zeigt, wie mit der Anweisung `.LS` einfacher und doppelter Zeilenabstand definiert wird. Es zeigt auch, wie diese Anweisung mit der Anweisung `.SL` zusammenwirkt.

Rohtext

```
.LM 5;.RM 55
.FI ON;.JU ON
Als Anfangswert ist der einzeilige Zeilenabstand aktiv (.LS S). Sie können
auch doppelten Zeilenabstand angeben (.LS D).
.SL 1
.LS D
Für diesen Abschnitt wurde doppelter Zeilenabstand definiert. Das bedeutet,
dass zwischen den Textzeilen jeweils eine Leerzeile ausgegeben wird.
.SL 1
Wenn Sie die Anweisung .SL mit doppeltem Zeilenabstand benutzen, wird der
Abstand zwischen den beiden Abschnitten erhöht, d.h. es werden 3 Leerzeilen
eingefügt.
.LS S
```

.SL 1

Nun wurde einzeliger Abstand VOR der Anweisung .SL definiert. Aus diesem Grund wird vor diesem Abschnitt eine einzige Leerzeile ausgegeben.

Formatierte Ausgabe

Als Anfangswert ist der einzelige Zeilenabstand aktiv (.LS S). Sie können auch doppelten Zeilenabstand angeben (.LS D).

Für diesen Abschnitt wurde doppelter Zeilenabstand definiert. Das bedeutet, dass zwischen den Textzeilen jeweils eine Leerzeile ausgegeben wird.

Wenn Sie die Anweisung .SL mit doppeltem Zeilenabstand benutzen, wird der Abstand zwischen den beiden Abschnitten erhöht, d.h. es werden 3 Leerzeilen eingefügt.

Nun wurde einzeliger Abstand VOR der Anweisung .SL definiert. Aus diesem Grund wird vor diesem Abschnitt eine einzige Leerzeile ausgegeben.

.CS - Schreibdichte

```
.CS zahl
```

Diese Funktion wird in Verbindung mit einem Con-nect-Druckerprofil benutzt (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Druckerprofile*). Sie müssen Ihre Definitionen für die Schreibdichte in das Druckerprofil aufnehmen, damit die Anweisung .CS die gewünschte Ausgabe liefert. Beispiel: Für die Anweisung .CS 100 müssen Sie Folgendes in Ihr Druckerprofil eingeben:

```
CI100
```

Sie können die Schreibdichte definieren, d.h. wie viele Zeichen pro Zoll in einer Zeile gedruckt werden sollen. Der Anfangswert ist 10 Zeichen pro Zoll, was der folgenden Anweisung entspricht:

```
.CS 100
```

Der Parameter *zahl* muss mit genau drei Stellen angegeben werden. Führende Nullen müssen immer angegeben werden (z.B.: .CS 060 für 6 Zeichen pro Zoll).

Einige Beispiele zum Definieren der Schreibdichte:

```
.CS 060
Diese Anweisung definiert 6 Zeichen pro Zoll.
Dies wurde mit der Anweisung .CS 060 erreicht.
.SL 1
.CS 100
Diese Anweisung definiert 10 Zeichen pro Zoll.
Dies wurde mit der Anweisung .CS 100 erreicht.
.SL 1
.CS 120
Diese Anweisung definiert 12 Zeichen pro Zoll.
Dies wurde mit der Anweisung .CS 120 erreicht.
.SL 1
.CS 150
Diese Anweisung definiert 15 Zeichen pro Zoll.
Dies wurde mit der Anweisung .CS 150 erreicht.
```

Zeilenabstand und Schreibdichte am Seitenanfang und Seitenende

Wenn in Ihrem Text viele .LS- und .CS-Anweisungen verwendet werden, sollten Sie separate .LS- und .CS-Werte für die Titel- und Fußzeilen definieren. Andernfalls kann es passieren, dass auf den verschiedenen Seiten ein anderer Zeilenabstand und eine andere Schreibdichte in den Titel- und Fußzeilen verwendet wird.

Für die folgenden Bereiche können Sie individuelle Werte für Zeilenabstand und Schreibdichte definieren:

- Titelzeile (.TT, .ET und .OT) einschließlich Leerzeilen (.TM und .HM)
- Fußzeile (.BT, .EB und .OB) einschließlich Leerzeilen (.FM)

Hierzu geben Sie die Anweisungen .LS und .CS direkt vor den Anweisungen ein, die die Titel- und Fußzeilen beeinflussen. Beispiel:

```
.LS 030;.CS 060
.TT Die Titelzeilen werden mit .LS 030 und .CS 060 gedruckt
.LS 040;.CS 100
.BT Die Fußzeilen werden mit .LS 040 und .CS 100 gedruckt
Der Dokumentrumpf wird auch mit .LS 040 und .CS 100 gedruckt.
.LS 060;.CS 120
Nun wird der Dokumentrumpf mit .LS 060 und .CS 120 gedruckt.
```

Im oben aufgeführten Beispiel werden unterschiedliche Werte für die Titelzeilen, die Fußzeilen und den Dokumentrumpf benutzt. Im folgenden Beispiel werden dieselben Werte für Titel- und Fußzeilen benutzt:

```
.LS 030;.CS 060
.TT Die Titelzeilen werden mit .LS 030 und .CS 060 gedruckt
.BT Die Fußzeilen werden auch mit .LS 030 und .CS 060 gedruckt
.LS 060;.CS 120
Der Dokumentrumpf wird mit .LS 060 und .CS 120 gedruckt.
Wenn Sie die Anweisungen .LS 060 und .CS 120 weglassen, werden
in den Titel- und Fußzeilen und im Dokumentrumpf dieselben
Werte benutzt (.LS 030 und .CS 060).
```

Die Anweisung .LS kann bewirken, dass die Zeilen am Seitenende unterschiedlich positioniert werden. Aus diesem Grund wird eine interne Berechnung durchgeführt und, wenn erforderlich, eine Leerzeile gedruckt. Für diese Leerzeile wird die passendste Zeilenabstandsdefinition aus dem Con-nect-Druckerprofil herangezogen. Daher wird empfohlen, im Con-nect-Druckerprofil so viele Zeilenabstandsdefinitionen wie möglich anzugeben (z.B. LI060), damit die Zeilen am Seitenende auf allen Seiten gleich positioniert werden können.

Siehe auch: *Titelzeile und Fußzeile*, *Leerzeilen am Seitenanfang* und *Leerzeilen am Seitenende*.

Zeilenabstand und Schreibdichte in Kopfzeilen

Mit der Anweisung `.HL` können für Kopfzeilen separate Werte für Zeilenabstand und Schreibdichte definiert werden. Beispiel:

```
.HL .LS 060  
.HL .CS 120  
.HL Dies ist die Kopfzeile.
```

Wenn Sie Zeilenabstand und Schreibdichte nicht mit der Anweisung `.HL` angeben, werden die Werte herangezogen, die gerade für den Dokumentrumpf benutzt werden. Es wird empfohlen, dass Sie den Zeilenabstand mit der Anweisung `.HL` definieren. Andernfalls kann das Layout der Kopfzeilen auf den verschiedenen Seiten unterschiedlich sein.

Ausführliche Informationen über die Anweisung `.HL` finden Sie unter [.HL - Kopfzeile](#).

Zeilenabstand und Schreibdichte in Fußnoten

Für Fußnoten können separate Werte für Zeilenabstand und Schreibdichte definiert werden. Hierzu müssen Sie die Anweisungen `.LS` und `.CS` wie in dem folgenden Beispiel in einem separaten Fußnotenblock definieren:

```
.FN ON  
.LS 060  
.CS 120  
.FN OFF
```

Die Werte in diesem Fußnotenblock gelten für alle Fußnoten, die im Rohtext folgen. Daher muss dieser Block vor der ersten Fußnote stehen, die ausgegeben werden soll.

Wenn Sie bei der nächsten Fußnote andere Werte verwenden wollen, müssen Sie vor dieser Fußnote einen neuen Fußnotenblock definieren. Die Werte in diesem Block gelten dann für alle nachfolgenden Fußnoten.



Anmerkung: Das Konzept eines separaten Fußnotenblocks für Zeilenabstand und Schreibdichte stellt eine Ausnahme dar. Um ein anderes Layout (z.B. linke und rechte Ränder) für die Fußnoten zu definieren, müssen Sie dies weiterhin für jede Fußnote separat angeben.

Ausführliche Informationen über Fußnoten finden Sie unter [.FN - Fußnote](#).

4 Text anordnen

- Text ausrichten 87
- Text einrücken 94
- Seitenumbruch 103
- Auf einer Seite verfügbare Zeilen 106

Dieses Kapitel behandelt die folgenden Themen:

Text ausrichten

Neben der Möglichkeit, Flattersatz (siehe [.FI - Zeilenausgleich](#)) oder Blocksatz zu definieren (siehe [.JU - Randausgleich](#)), ist es auch möglich, den Text rechtsbündig oder zentriert anzuordnen.

.CE - Text zentrieren

```
.CE zahl  
.CE ON  
.CE OFF
```

Mit dieser Anweisung können Sie eine Textzeile zwischen dem linken und rechten Rand zentrieren. Solange diese Anweisung wirksam ist, werden Zeilenausgleich (.FI) und Randausgleich (.JU) nicht ausgeführt.

Das Zentrieren kann z.B. mit Fettdruck (siehe [.BF - Fettdruck](#)) oder Unterstreichung (siehe [.UL - Unterstreichen](#)) kombiniert werden.

Die Anweisung .CE kann auf zwei Arten benutzt werden:

Eine bestimmte Anzahl von Zeilen zentrieren

Geben Sie an, wie viele Zeilen zentriert werden sollen. Um z.B. die nächsten beiden Zeilen zu zentrieren, geben Sie ein:

```
.CE 2
```

Einen Textblock zentrieren

Sie können auch einen Textblock zentrieren. Dazu müssen Sie den Text zwischen den beiden folgenden Anweisungen eingeben:

```
.CE ON
```

```
.CE OFF
```

Beispiel

Dieses Beispiel zeigt, wie Text zentriert wird. Es enthält außerdem die Anweisungen **.BP** für Fettdruck und **.UL** zum Unterstreichen.

Rohtext

```
.LM 0;.RM 60
.FI ON;.JU ON
.CE 4
Dies ist zentrierter Text. Da Zeilenausgleich
und Randausgleich dafür nicht gelten, erscheint jede
Zeile des Rohtextes als eine Zeile in der
formatierten Ausgabe.
.SL 1
Im obigen Absatz wurden vier Zeilen des Textes zentriert.
Der folgende Text wird zwischen dem linken und dem rechten Rand
angeordnet (wie mit den Anweisungen .LM und .RM definiert).
.SL 1
.CE ON
.** Dies ist ein Beispiel für Fettdruck im zentrierten Text.
.BP
Sie können auch einen Textblock zentrieren.
Dazu müssen Sie den Text zwischen die Anweisungen .CE ON
und .CE OFF schreiben.
.SL 1
Sie müssen beim Zentrieren darauf achten,
dass jede Zeile zwischen den definierten
linken und rechten Rand passt.
.** Dies ist ein Beispiel für unterstrichenen zentrierten Text.
.** Die folgende Zeile wird unterstrichen. Beachten Sie, dass die
.** Anweisung .UL NACH dem zu unterstreichenden Text steht.
Andernfalls kann die Zeile nicht richtig formatiert werden.
.UL *
.CE OFF
.SL 1
Dieser Text folgt dem zentrierten Textblock. Er wird zwischen dem linken
Rand (mit .LM definiert) und dem rechten Rand (mit .RM definiert)
angeordnet.
```

Formatierte Ausgabe

Dies ist zentrierter Text. Da Zeilenausgleich
und Randausgleich dafür nicht gelten, erscheint jede
Zeile des Rohtextes als eine Zeile in der
formatierten Ausgabe.

Im obigen Absatz wurden vier Zeilen des Textes zentriert.
Der folgende Text wird zwischen dem linken und dem rechten
Rand angeordnet (wie mit den Anweisungen .LM und .RM
definiert).

Sie können auch einen Textblock zentrieren.
Dazu müssen Sie den Text zwischen die Anweisungen .CE ON

und .CE OFF schreiben.

Sie müssen beim Zentrieren darauf achten,
dass jede Zeile zwischen den definierten
linken und rechten Rand passt.

Andernfalls kann die Zeile nicht richtig formatiert werden.

Dieser Text folgt dem zentrierten Textblock. Er wird
zwischen dem linken Rand (mit .LM definiert) und dem rechten
Rand (mit .RM definiert) angeordnet.

.RA - Rechtsbündige Textausgabe

```
.RA zah1  
.RA ON  
.RA OFF
```

Mit dieser Anweisung richten Sie den Text so aus, dass der rechte Rand gerade und der linke Rand als Flatterrand ausgegeben wird. Solange diese Anweisung wirksam ist, werden der Zeilenausgleich (.FI) und der Randausgleich (.JU) nicht ausgeführt.

Die rechtsbündige Textausgabe kann z.B. mit Fettdruck (siehe **.BF - Fettdruck**) oder Unterstreichung (siehe **.UL - Unterstreichen**) kombiniert werden.

Die Anweisung .RA kann auf zwei Arten benutzt werden:

Text an den definierten rechten Rand verschieben

Sie können den Text nach rechts bis zum rechten Rand verschieben, der mit der Anweisung .RM definiert wurde. Dazu müssen Sie den Text zwischen die beiden folgenden Anweisungen schreiben:

```
.RA ON
```

```
.RA OFF
```

Den rechten Rand ändern

Sie können für rechtsbündig anzuordnenden Text auch einen anderen rechten Rand definieren. Wenn der rechte Rand z.B. in Spalte 50 sein soll, müssen Sie den Text zwischen die beiden folgenden Anweisungen schreiben:

```
.RA 50
```

```
.RA OFF
```

Beispiel

Dieses Beispiel zeigt die rechtsbündige Textausgabe. Es enthält auch die Anweisungen **.BF** (Fett-**druck**) und **.US** (Unterstreichen). Wenn Sie dieses Beispiel mit dem vorhergehenden **Beispiel zum Zentrieren von Text** vergleichen, werden Sie feststellen, dass dort andere Anweisungen benutzt wurden (.BP und .UL).

Rohtext

```
.LM 0;.RM 60
.FI ON;.JU ON
.RA ON
.** Dies ist ein Beispiel für Fettdruck bei rechtsbündigem Text.
.BF 1
Dies ist rechtsbündiger Text.
.BF
Da Zeilenausgleich und Randausgleich hier nicht
gelten, erscheint jede Zeile des Rohtextes
als eine Zeile in der formatierten Ausgabe.
.RA OFF
.SL 1
Dieser Textabschnitt wird zwischen dem linken und dem rechten Rand
angeordnet (wie mit den Anweisungen .LM und .RM definiert).
.SL 1
.RA 40
Der rechte Rand für diesen
Teil des rechtsbündigen
Textes wurde in
.** Dies ist ein Beispiel für unterstrichenen rechtsbündigen Text.
.** Anders als die Anweisung .UL muss die in diesem Beispiel benutzte
.** Anweisung .US VOR dem Text angegeben werden, der rechtsbündig
.** angeordnet werden soll.
.US
Spalte 40 definiert.
.RA OFF
.SL 1
Ab hier wird der Text wieder zwischen dem linken und rechten Rand
gleichmäßig ausgerichtet.
```

Formatierte Ausgabe

```
                Dies ist rechtsbündiger Text.
                Da Zeilenausgleich und Randausgleich hier nicht
                gelten, erscheint jede Zeile des Rohtextes
                als eine Zeile in der formatierten Ausgabe.

Dieser Textabschnitt wird zwischen dem linken und dem
rechten Rand angeordnet (wie mit den Anweisungen .LM und .RM
definiert).
```


Der rechte Rand für diesen
Teil des rechtsbündigen
Textes wurde in
Spalte 40 definiert.

Ab hier wird der Text wieder zwischen dem linken und rechten
Rand gleichmäßig ausgerichtet.

Text einrücken

Sie können für die erste Zeile eines Abschnitts definieren, dass sie in einer anderen Spalte beginnen soll als die restlichen Zeilen des Abschnitts.

Die Anweisungen `.TI` und `.OF` können nicht beide im gleichen Abschnitt benutzt werden.

.TI - Einzug einer Zeile

```
.TI zahl  
.TI +zahl  
.TI -zahl
```

Die Anweisung `.TI` definiert den linken Rand für die erste Zeile. Alle folgenden Zeilen benutzen den linken Rand, der mit der Anweisung `.LM` definiert wurde. Die Anweisung `.TI` bewirkt eine Unterbrechung des Zeilenausgleichs.

Die Variable `$IN` enthält den aktuellen Wert für den Einzug der ersten Zeile (siehe [Veränderbare Systemvariablen](#)).

Die Anweisung `.TI` kann auf drei verschiedene Arten benutzt werden:

Eine bestimmte Spalte definieren

Sie können die Spalte angeben, in der die erste Zeile beginnen soll, z.B. Spalte 5 (während die folgenden Zeilen in Spalte 0 beginnen):

```
.TI 5
```

Nach rechts verschieben

Beispiel: Der aktuelle linke Rand ist Spalte 10. Um die erste Zeile 3 Spalten nach rechts zu verschieben, so dass sie in Spalte 13 beginnt, geben Sie an:

```
.TI +3
```

Nach links verschieben

Beispiel: Der aktuelle linke Rand ist Spalte 10. Um die erste Zeile 4 Spalten nach links zu verschieben, so dass sie in Spalte 6 beginnt, geben Sie an:

```
.TI -4
```

Der absolute linke Rand kann nicht über Spalte 0 hinausgehen, d.h. in diesem Beispiel wäre es nicht möglich, als Parameter -11 anzugeben (weil 10 als linker Rand definiert ist).

Beispiel

Dieses Beispiel zeigt die Anweisung .TI zum Einrücken der ersten Zeile eines Abschnitts.

Rohtext

```
.LM 5;.RM 60  
.FI ON;.JU ON  
.TI 20
```

Die erste Zeile dieses Abschnitts beginnt in Spalte 20. Alle folgenden Zeilen beginnen am definierten linken Rand, d.h. in Spalte 5. Die erste Zeile wird 15 Zeichen nach rechts eingerückt, so dass sie in Spalte 20 anfängt.

```
.IL 1  
.TI +15
```

Die erste Zeile dieses Abschnitts beginnt in der gleichen Spalte wie die erste Zeile des vorigen Abschnitts.

```
.IL 1  
.TI -3
```

Wenn der linke Rand größer als 0 ist, können Sie die erste Zeile auch nach links verschieben. Der dafür benutzte Wert darf aber nicht größer sein als der für den linken Rand benutzte Wert, da der Text links nicht über die Position 0 hinaus verschoben werden kann.

Formatierte Ausgabe

Die erste Zeile dieses Abschnitts beginnt in Spalte 20. Alle folgenden Zeilen beginnen am definierten linken Rand, d.h. in Spalte 5. Die erste Zeile wird 15 Zeichen nach rechts eingerückt, so dass sie in Spalte 20 anfängt.

Die erste Zeile dieses Abschnitts beginnt in der gleichen Spalte wie die erste Zeile des vorigen Abschnitts.

Wenn der linke Rand größer als 0 ist, können Sie die erste Zeile auch nach links verschieben. Der dafür benutzte Wert darf aber nicht größer sein als der für den linken Rand benutzte Wert, da der Text links nicht über die Position 0 hinaus verschoben werden kann.

.OF - Einzug mehrerer Zeilen

```
.OF zah1
.OF +zah1
.OF -zah1
.OF
```

Die erste Zeile des Abschnitts benutzt den linken Rand, der mit der Anweisung `.LM` definiert wurde. Die Anweisung `.OF` bestimmt den linken Rand für alle weiteren Zeilen. Sie bewirkt eine Unterbrechung des Zeilenausgleichs.

Die Variable `$IN` enthält den aktuellen Wert für den Einzug mehrerer Zeilen (siehe [Veränderbare Systemvariablen](#)).

Die Anweisung `.OF` kann auf vier verschiedene Arten benutzt werden:

Eine bestimmte Spalte definieren

Sie können angeben, in welcher Spalte alle Zeilen nach der ersten Zeile beginnen sollen. Wenn die erste Zeile z.B. in Spalte 0 beginnt und alle folgenden Zeilen in Spalte 5 beginnen sollen, geben Sie an:

```
.OF 5
```

Verschieben nach rechts

Wenn die erste Zeile z.B. in Spalte 10 beginnt und die folgenden Zeilen 6 Spalten nach rechts verschoben werden sollen, so dass sie in Spalte 16 beginnen, geben Sie an:

```
.OF +6
```

Verschieben nach links

Wenn die erste Zeile z.B. in Spalte 10 beginnt und die folgenden Zeilen 4 Spalten nach links verschoben werden sollen, so dass sie in Spalte 6 beginnen, geben Sie an:

```
.OF -4
```

Der absolute linke Rand kann nicht über Spalte 0 hinausgehen, d.h. in diesem Beispiel wäre es nicht möglich, als Parameter `-11` anzugeben (weil 10 als linker Rand definiert ist).

Die Anweisung .OF rückgängig machen

Jede .OF-Anweisung definiert einen neuen Wert und macht automatisch die vorherige .OF-Anweisung unwirksam.

Wenn Sie die Anweisung .OF rückgängig machen möchten, so dass wieder der mit der Anweisung .LM definierte linke Rand gilt, müssen Sie die Anweisung ohne Parameter angeben:

```
.OF
```

Der mit der Anweisung .OF definierte Wert wird auch dann unwirksam, wenn Sie mit der Anweisung .LM einen neuen linken Rand definieren.

Beispiel

Dieses Beispiel zeigt den Einzug mehrerer Zeilen mit der Anweisung `.OF`.

Rohtext

```
.LM 5;.RM 60
.FI ON;.JU ON
.OF 9
1. Sie können die Anweisung .OF verwenden, um z.B. eine nummerierte
Liste zu erstellen.
.SL 1;.OF 9
2. Anders als die Anweisung .TI gilt die Anweisung .OF für alle Zeilen
nach der ersten Zeile.
.SL 1;.OF 9
3. Die erste Zeile ist von der Anweisung .OF nicht betroffen.
Sie beginnt am linken Rand.
.SL 1;.OF +4
4. Die Zeilen dieses Abschnitts (außer der ersten) beginnen in der gleichen
Spalte wie die Zeilen des vorigen Abschnitts.
.SL 1;.OF -5
5. Sie können auch negative Werte für den Einzug angeben. In diesem
Beispiel beginnt die erste Zeile am linken Rand (also in Spalte 5), und
die übrigen Zeilen beginnen in Spalte 0.
.SL 1;.OF
Die Anweisung .OF ohne Parameter schaltet den Einzug aus, und alle Zeilen
beginnen am linken Rand wie mit der Anweisung .LM definiert.
```

Formatierte Ausgabe

```
1. Sie können die Anweisung .OF verwenden, um z.B.
eine nummerierte Liste zu erstellen.

2. Anders als die Anweisung .TI gilt die Anweisung .OF
für alle Zeilen nach der ersten Zeile.

3. Die erste Zeile ist von der Anweisung .OF nicht
betroffen. Sie beginnt am linken Rand.

4. Die Zeilen dieses Abschnitts (außer der ersten)
beginnen in der gleichen Spalte wie die Zeilen des
vorigen Abschnitts.

5. Sie können auch negative Werte für den Einzug
angeben. In diesem Beispiel beginnt die erste Zeile am
linken Rand (also in Spalte 5), und die übrigen Zeilen
beginnen in Spalte 0.

Die Anweisung .OF ohne Parameter schaltet den Einzug
```

aus, und alle Zeilen beginnen am linken Rand wie mit der Anweisung `.LM` definiert.

.PS - Absatz

.PS

Mit der Anweisung .PS definieren Sie einen Absatz. Die erste Zeile eines Absatzes wird automatisch drei Spalten nach rechts eingezogen.

Die Anweisung .PS führt automatisch die folgenden vier Anweisungen aus:

.BR

Der Zeilenausgleich wird unterbrochen.

.NL 4

Wenn auf der aktuellen Seite weniger als 4 Zeilen zur Verfügung stehen, beginnt der neue Absatz auf der nächsten Seite (siehe *.NL - Benötigte Zeilen*).

.SL

Diese Anweisung wirkt mit dem Zeilenabstand (.LS) zusammen.

Bei einfachem Zeilenabstand wird zwischen zwei Absätzen eine Zeile eingefügt (sofern kein Seitenumbruch erfolgt).

Bei doppeltem Zeilenabstand werden drei Zeilen zwischen zwei Absätzen eingefügt (sofern kein Seitenumbruch erfolgt).

.TI +3

Die erste Zeile des Absatzes wird drei Spalten nach rechts eingezogen.

Beispiel

Dieses Beispiel zeigt die Anweisung .PS zusammen mit doppeltem Zeilenabstand.

Rohtext

```
.LM 5;.RM 55  
.FI ON;.JU ON  
.LS D;.PS  
Für diesen Abschnitt wurde doppelter Zeilenabstand definiert, d.h.  
zwischen den Textzeilen wird jeweils eine Leerzeile ausgegeben.  
.PS  
Wenn der doppelte Zeilenabstand eingeschaltet ist, werden zwischen den  
Abschnitten drei Leerzeilen ausgegeben (anstelle einer Leerzeile beim  
einfachen Zeilenabstand).
```

Formatierte Ausgabe

Für diesen Abschnitt wurde doppelter
Zeilenabstand definiert, d.h. zwischen den
Textzeilen wird jeweils eine Leerzeile ausgegeben.

Wenn der doppelte Zeilenabstand eingeschaltet
ist, werden zwischen den Abschnitten drei
Leerzeilen ausgegeben (anstelle einer Leerzeile
beim einfachen Zeilenabstand).

Seitenumbruch

Die folgenden Anweisungen steuern die Seitenumbrüche. Wenn Sie diese Anweisungen nicht verwenden, gelten die Anfangswerte für das Dokumentlayout; ein Seitenumbruch erfolgt demnach nach der Fußzeile. Falls keine Fußzeile definiert ist, erfolgt der Seitenumbruch nach der Leerzeile, die anfänglich für die Fußzeile reserviert ist.

.NP - Neue Seite

```
.NP zahl  
.NP
```

Die folgende Anweisung bewirkt einen Seitenumbruch:

```
.NP
```

Vor dem Seitenumbruch druckt Con-form die definierten Fußnoten und die Fußzeile. Der nach der Anweisung .NP stehende Text beginnt auf der nächsten Seite.

Wenn der zu druckende Text bereits am Seitenanfang angeordnet wurde, findet kein Seitenumbruch statt. Wenn die Anweisung .NP beispielsweise ganz am Anfang des Textes steht, wird der Seitenumbruch nicht ausgeführt.

Sie können eine neue Seitenzahl für die nächste Seite definieren. Wenn die neue Seite z.B. die Seitenzahl 6 haben soll, geben Sie an:

```
.NP 6
```

Die Anweisung .NP bewirkt lediglich einen Seitenumbruch. Wenn Sie mehrere Leerseiten definieren möchten, müssen Sie die Anweisung .IP verwenden.

Die Variable \$PN enthält die aktuelle Seitenzahl (siehe [Veränderbare Systemvariablen](#)).

.IP - Seiten einfügen

```
.IP zah1  
.IP
```

Diese Anweisung fügt Leerseiten in den Ausgabebetext ein. Falls die Titelzeile, Kopfzeile und Fußzeile definiert sind, werden sie auch auf den Leerseiten ausgedruckt.

Beispiel: Um 3 Leerseiten einzufügen, geben Sie an:

```
.IP 3
```

Falls diese Anweisung in der Mitte einer Seite ausgeführt wird, bleibt die restliche Seite leer und die angegebenen Leerseiten werden eingefügt. Der auf die Anweisung .IP folgende Text erscheint am Anfang der Seite, die nach den Leerseiten ausgegeben wird.

Wenn Sie diese Anweisung ohne Parameter eingeben, wird eine Leerseite eingefügt.

.OP PGF - Seitenformatierung

```
.OP PGF=ON  
.OP PGF=OFF
```

Sie können die Seitenformatierung ausschalten, so dass Seitenumbrüche unterbleiben. In diesem Fall wird nur der Dokumentrumpf ausgedruckt. Anstatt auf einer neuen Seite wird der Text in einer neuen Zeile ausgegeben. Dies ist z.B. sinnvoll, wenn Sie nur den eigentlichen Text zum Korrekturlesen drucken möchten.

Bei ausgeschalteter Seitenformatierung wird Folgendes nicht ausgegeben: Titelzeile, Fußzeile, Kopfzeile und Fußnoten. Einträge für das Inhaltsverzeichnis, Indexeinträge sowie Textblöcke, die mit den Anweisungen .NL KEEP und .NL FLOAT definiert wurden, werden ebenfalls ignoriert.

Bei ausgeschalteter Seitenformatierung bleiben andere Anweisungen für das Dokumentlayout, z.B. der linke und der rechte Rand, Zeilenausgleich, Randausgleich, zentrierter Text und Leerzeilen weiterhin wirksam.

Um die Seitenformatierung für das gesamte Dokument auszuschalten, geben Sie die folgende Anweisung am Anfang des Dokuments ein:

```
.OP PGF=OFF
```

Die Seitenformatierung ist anfänglich eingeschaltet. Dies entspricht der Anweisung:

```
.OP PGF=ON
```

Auf einer Seite verfügbare Zeilen

.NL - Benötigte Zeilen

```
.NL zahl  
.NL  
.NL KEEP  
.NL FLOAT  
.NL OFF
```

Die Anweisung .NL ist sinnvoll, wenn Sie sicherstellen möchten, dass ein Absatz nicht kurz vor dem unteren Rand einer Seite beginnt, oder wenn Sie für eine Abbildung Platz freilassen möchten, der nicht durch einen Seitenumbruch geteilt werden soll.

Diese Anweisung stellt fest, ob die benötigten Zeilen auf der aktuellen Ausgabeseite verfügbar sind. Wenn nicht, wird der Text auf der nächsten Seite ausgegeben. Wenn die benötigten Zeilen verfügbar sind, bewirkt die Anweisung .NL eine Unterbrechung des Zeilenausgleichs, d.h. der Text fährt in einer neuen Zeile fort.

Die Variable \$LC enthält die Anzahl der verbleibenden Zeilen auf der aktuellen Seite (siehe [Veränderbare Systemvariablen](#)).

Im Folgenden werden die verschiedenen Formen der Anweisung .NL beschrieben.

.NL zahl

Wenn Sie z.B. sicherstellen möchten, dass die folgenden 8 Zeilen zusammenhängend ausgegeben werden, geben Sie an:

```
.NL 8
```

Wenn auf der aktuellen Seite noch 8 Zeilen verfügbar sind, beginnt der Text in der nächsten Zeile. Wenn nicht, wird der Text auf der nächsten Seite ausgegeben.

Wenn Sie Leerzeilen reservieren möchten (z.B. für eine Abbildung, die später der gedruckten Seite hinzugefügt werden soll), geben Sie außerdem die Anweisung .IL an. Beispiel:

```
.NL 10;.IL 10
```

Die Anweisung `.NL` ohne Parameter entspricht der Anweisung `.NL 7`.

.NL OFF

Diese Anweisung definiert das Ende eines Textblocks. Sie wird in Zusammenhang mit den Anweisungen `.NL KEEP` und `.NL FLOAT` benutzt, die beide im Folgenden beschrieben sind.

Mit einer **.IF-Anweisung** ist es nicht möglich, die Verarbeitung von `.NL OFF` bedingt zu unterdrücken.

.NL KEEP

Sie können einen Textblock zwischen die beiden folgenden Anweisungen schreiben:

```
.NL KEEP
```

```
.NL OFF
```

Con-form prüft dann automatisch, ob die für diesen Textblock benötigten Zeilen auf der aktuellen Seite verfügbar sind. Wenn ja, erscheint der Textblock auf der aktuellen Seite; wenn nicht, wird er auf der nächsten Seite gedruckt.

Beispiel

Dieses Beispiel zeigt, wie Leerzeilen für eine Abbildung reserviert werden und wie sich die Anweisung `.NL KEEP` auswirkt.

Rohtext

```
.TT Con-form Benutzerhandbuch//Der KEEP-Puffer
.BT //Seite #
.LM 10;.RM 60
.FI ON;.JU ON
```

Sie können Platz reservieren für eine Abbildung, die später der gedruckten Ausgabe hinzugefügt wird. Auf dieser Seite wurden 30 Leerzeilen reserviert.

```
.IL 1
```

Für den Fall, dass auf der Seite weniger als 30 Zeilen verfügbar sind, werden die Leerzeilen automatisch am Anfang der nächsten Seite eingefügt.

```
.NL 30;.IL 30
```

Alle Leerzeilen vor dieser Zeile wurden für eine Abbildung reserviert. Wenn der Text im folgenden KEEP-Puffer länger ist als die verbleibende Seite, wird er automatisch auf der nächsten Seite gedruckt. Andernfalls

wird er auf der aktuellen Seite ausgegeben.

```
.IL 2
```

```
.NL KEEP
```

Die Anweisung `.NL KEEP`

```
.SL 1
```

Mit der Anweisung `.NL KEEP` können Sie einen Textabschnitt in einem Puffer speichern. Wenn auf der aktuellen Seite genügend Zeilen für den im Puffer stehenden Text verfügbar sind, wird der Text auf der aktuellen Seite gedruckt.

```
.SL 1
```

Wenn auf der Seite nicht genügend Platz frei ist, wird der Text automatisch auf der nächsten Seite gedruckt.

```
.NL OFF
```


Erste Seite der formatierten Ausgabe

Con-form Benutzerhandbuch	Der KEEP-Puffer
<p>Sie können Platz reservieren für eine Abbildung, die später der gedruckten Ausgabe hinzugefügt wird. Auf dieser Seite wurden 30 Leerzeilen reserviert.</p> <p>Für den Fall, dass auf der Seite weniger als 30 Zeilen verfügbar sind, werden die Leerzeilen automatisch am Anfang der nächsten Seite eingefügt.</p>	
<p>Alle Leerzeilen vor dieser Zeile wurden für eine Abbildung reserviert. Wenn der Text im folgenden KEEP-Puffer länger ist als die verbleibende Seite, wird er automatisch auf der nächsten Seite gedruckt. Andernfalls wird er auf der aktuellen Seite ausgegeben.</p>	
Seite 1	

Zweite Seite der formatierten Ausgabe

Con-form Benutzerhandbuch	Der KEEP-Puffer
Die Anweisung .NL KEEP	
Mit der Anweisung .NL KEEP können Sie einen Textabschnitt in einem Puffer speichern. Wenn auf der aktuellen Seite genügend Zeilen für den im Puffer stehenden Text verfügbar sind, wird der Text auf der aktuellen Seite gedruckt.	
Wenn auf der Seite nicht genügend Platz frei ist, wird der Text automatisch auf der nächsten Seite gedruckt.	
Seite 2	

.NL FLOAT

Zwischen den beiden folgenden Anweisungen können Sie einen Textblock angeben:

```
.NL FLOAT
```

```
.NL OFF
```

Con-form prüft dann, ob die für den Textblock benötigten Zeilen auf der aktuellen Seite verfügbar sind. Wenn ja, erscheint der Textblock auf der aktuellen Seite.

Wenn der Textblock nicht auf die aktuelle Seite passt, verarbeitet die Anweisung `.NL FLOAT` (anders als `.NL KEEP`) den Text, der der Anweisung `.NL OFF` folgt, bis die aktuelle Seite voll ist. Der definierte Textblock erscheint dann auf der nächsten Seite.

Die Anweisung `.NL FLOAT` ist geeignet für einen Textblock, dessen Position nicht beibehalten werden muss, so dass er sowohl auf der aktuellen als auch auf der nächsten Seite ausgegeben werden kann.

Beispiel

Dieses Beispiel reserviert Leerzeilen für eine Abbildung. Es zeigt außerdem die Auswirkungen der Anweisung `.NL FLOAT`.

Rohtext

```
.TT Con-form Benutzerhandbuch//Der FLOAT-Puffer
.BT //Seite #
.LM 10;.RM 60
.FI ON;.JU ON
Sie können Leerzeilen für eine Abbildung reservieren, die später auf der
gedruckten Seite eingefügt wird. Auf dieser Seite wurden 30 Zeilen
reserviert.
.NL 30;.IL 30
Alle Leerzeilen über dieser Zeile wurden für eine Abbildung reserviert.
.IL 2
.NL FLOAT
Die Anweisung .NL FLOAT
.SL 1
Mit der Anweisung .NL FLOAT können Sie einen Textabschnitt in einem Puffer
speichern. Wenn auf der aktuellen Seite genügend Zeilen für den
Textabschnitt frei sind, wird der Text auf dieser Seite gedruckt.
.SL 1
Wenn auf der aktuellen Seite nicht genug Platz ist für den im Puffer
gespeicherten Text, wird die Seite mit dem Text gefüllt, der auf die
Anweisung .NL OFF folgt. Der Text im Puffer wird oben auf der nächsten
```

Seite gedruckt.

```
.SL 1
```

Die Anweisung `.NL FLOAT` ist also geeignet für Text, der in der formatierten Ausgabe nicht unbedingt an der Stelle stehen muss, an der er im Rohtext eingegeben wurde.

```
.SL 1
```

```
.NL OFF
```

Dieser Text wurde im Rohtext nach der Anweisung `.NL OFF` eingegeben. Er wird vor dem im `FLOAT`-Puffer gespeicherten Text ausgedruckt, falls auf der aktuellen Seite nicht genügend Zeilen für den Text im Puffer vorhanden sind.

Erste Seite der formatierten Ausgabe

Con-form Benutzerhandbuch	Der FLOAT-Puffer
<p>Sie können Leerzeilen für eine Abbildung reservieren, die später auf der gedruckten Seite eingefügt wird. Auf dieser Seite wurden 30 Zeilen reserviert.</p>	
<p>Alle Leerzeilen über dieser Zeile wurden für eine Abbildung reserviert.</p>	
<p>Dieser Text wurde im Rohtext nach der Anweisung .NL OFF eingegeben. Er wird vor dem im FLOAT-Puffer gespeicherten Text ausgedruckt, falls auf der aktuellen Seite nicht genügend Zeilen für den Text im Puffer vorhanden sind.</p>	
Seite 1	

Zweite Seite der formatierten Ausgabe

Con-form Benutzerhandbuch	Der FLOAT-Puffer
Die Anweisung .NL FLOAT	
Mit der Anweisung .NL FLOAT können Sie einen Textabschnitt in einem Puffer speichern. Wenn auf der aktuellen Seite genügend Zeilen für den Textabschnitt frei sind, wird der Text auf dieser Seite gedruckt.	
Wenn auf der aktuellen Seite nicht genug Platz ist für den im Puffer gespeicherten Text, wird die Seite mit dem Text gefüllt, der auf die Anweisung .NL OFF folgt. Der Text im Puffer wird oben auf der nächsten Seite gedruckt.	
Die Anweisung .NL FLOAT ist also geeignet für Text, der in der formatierten Ausgabe nicht unbedingt an der Stelle stehen muss, an der er im Rohtext eingegeben wurde.	
Seite 2	

.OP BRN - Kein Zeilenumbruch bei .NL

```
.OP BRN=ON  
.OP BRN=OFF
```

Normalerweise bewirkt die Anweisung `.NL` eine Unterbrechung des Zeilenausgleichs (d.h. einen Zeilenumbruch), wenn die benötigten Zeilen auf der aktuellen Seite verfügbar sind. Dies kann auch ausgeschaltet werden, so dass der Zeilenausgleich fortgesetzt wird.

Um den Zeilenumbruch auszuschalten, geben Sie an:

```
.OP BRN=OFF
```

Die anfängliche Einstellung dieser Option ist:

```
.OP BRN=ON
```

Beispiel

Dieses Beispiel zeigt die Auswirkung der Anweisung `.OP BRN=OFF`.

Rohtext

```
.LM 0;.RM 60  
.FI ON;.JU ON  
.OP BRN=OFF
```

Normalerweise bewirkt die Anweisung `.NL` eine Unterbrechung des Zeilenausgleichs. Diese Funktion kann ausgeschaltet werden.

```
.SL 1
```

In diesem Beispiel wurde die Option BRN ausgeschaltet. Sie können prüfen, ob die benötigten Zeilen auf der aktuellen Seite verfügbar sind.

```
.NL 8
```

Wenn genügend Zeilen vorhanden sind, wird der Zeilenausgleich NICHT unterbrochen.

Formatierte Ausgabe

Normalerweise bewirkt die Anweisung `.NL` eine Unterbrechung des Zeilenausgleichs. Diese Funktion kann ausgeschaltet werden.

In diesem Beispiel wurde die Option BRN ausgeschaltet. Sie können prüfen, ob die benötigten Zeilen auf der aktuellen

Seite verfügbar sind. Wenn genügend Zeilen vorhanden sind, wird der Zeilenausgleich NICHT unterbrochen.

5 Text hervorheben

▪ Fettdruck	119
▪ Text unterstreichen	124
▪ Kursivdruck	129
▪ Text hochstellen	130
▪ Text tiefstellen	131
▪ Text überdrucken	132
▪ Umschaltanweisungen	133

Dieses Kapitel behandelt die folgenden Themen:

Die Symbole, die in Umschaltanweisungen benutzt werden, repräsentieren Druckeranweisungen, die vorher gemäß dem entsprechenden Druckerhandbuch definiert wurden. Die folgenden Funktionen sind nur mit Hilfe von Umschaltanweisungen möglich: Kursivdruck, Text hochstellen, Text tiefstellen und Text überdrucken.

Fettdruck

Für Fettdruck stehen zwei Anweisungen zur Verfügung: `.BF` und `.BP`. Außer diesen Anweisungen können Sie auch Umschaltanweisungen benutzen, um auf Fettdruck umzuschalten.

Text, der mit den Anweisungen `.BF` und `.BP` fett gedruckt wird, kann auch zentriert (siehe [.CE - Text zentrieren](#)), rechtsbündig angeordnet (siehe [.RA - Rechtsbündige Textausgabe](#)) und unterstrichen werden (siehe [.UL - Unterstreichen](#)). Die zweite Art der Unterstreichung (siehe die Anweisung [.US](#)) ist in Kombination mit Fettdruck nur mit Umschaltanweisungen möglich.

.BF - Fettdruck

```
.BF zahl
.BF
```

Diese Anweisung bewirkt, dass der folgende Text fett gedruckt wird, bis die nächste `.BF`-Anweisung ausgeführt wird.

Sie sollten die Anweisung `.BF` verwenden, um größere Textabschnitte mit Fettdruck auszugeben. Die Anweisung `.BF` bewirkt einen Zeilenumbruch. Sie kann also nicht benutzt werden, um einzelne Wörter in einer Zeile fett zu drucken.

Um den Fettdruck einzuschalten, geben Sie an:

```
.BF 1
```



Anmerkung: Der Wert des Parameters `zahl` muss größer als 0 sein.

Der Text, der der Anweisung `.BF` folgt, wird fett gedruckt.

Um den Fettdruck auszuschalten, geben Sie die Anweisung `.BF` nochmals an, jedoch ohne Parameter:

```
.BF
```

.BP - Fettdruck

```
.BP
```

Diese Anweisung bewirkt Fettdruck für den Text, der im Rohtext in der Zeile nach der Anweisung steht.

Sie sollten die Anweisung .BP verwenden, um einzelne Wörter fett zu drucken. Diese Anweisung bewirkt keinen Zeilenumbruch. Es ist nicht möglich, einzelne Buchstaben innerhalb eines Wortes mit dieser Anweisung fett zu drucken. Dies ist nur mit Umschaltanweisungen möglich.

Um die auf die Anweisung folgende Textzeile fett zu drucken, geben Sie an:

```
.BP
```

Die Anweisung .BP muss nicht ausgeschaltet werden (im Gegensatz zu .BF).

Umschaltanweisungen für Fettdruck

Wenn Sie Umschaltanweisungen verwenden, können Sie auch Teile eines Wortes fett drucken sowie Fettdruck und Unterstreichung kombinieren (siehe [Umschaltanweisungen zum Unterstreichen von Text](#)). Umschaltanweisungen bewirken keinen Zeilenumbruch.

Bevor Sie eine Umschaltanweisung ausführen können, müssen Sie das Umschaltzeichen definieren (siehe [.OP ESC - Umschaltzeichen definieren](#)).

Um ein Wort teilweise fett zu drucken, müssen Sie im Rohtext Folgendes angeben:

M1	Fettdruck einschalten.
M0	Fettdruck ausschalten.

Das folgende Beispiel zeigt, wie ein Teil eines Wortes fett gedruckt wird:

```
.OP ESC=/  
Dies soll im /M1Fett/M0druck ausgegeben werden.
```

Die gedruckte Ausgabe sieht folgendermaßen aus:

Dies soll im **Fett**druck ausgegeben werden.

Beispiel

Dieses Beispiel enthält die Anweisungen .BF und .BP und zeigt, wie Umschaltanweisungen benutzt werden.

Rohtext

```
.LM 0;.RM 60
.FI ON;.JU ON
.BF 1
Der gesamte Text dieses Abschnitts wird fett gedruckt. Um den Fettdruck
auszuschalten, muss die Anweisung .BF nochmals eingegeben werden, diesmal
jedoch ohne Parameter.
.BF
.SL 1
In diesem Satz wird ein
.BP
einziges
Wort fett gedruckt. Dieses Wort muss direkt nach der Anweisung .BP
eingegeben werden, und es muss in einer
.BP
extra Zeile
stehen.
.SL 1
Wenn Sie Umschaltanweisungen benutzen möchten, müssen Sie zuerst das
Umschaltzeichen definieren.
.OP ESC=/
In diesem Beispiel ist der Schrägstrich das Umschaltzeichen.
Sie können nun /M1mehrere Wörter/M0 oder einen Teil eines Wortes im
/M1Fett/M0druck ausgeben.
.SL 1
.** Die folgende Anweisung macht die Definition des Umschaltzeichens
.** wieder rückgängig.
.OP ESC=NULL
Wenn Sie den Schrägstrich (/) im Text ausgeben möchten, müssen Sie das
Umschaltzeichen aufheben oder ein anderes definieren.
```

Formatierte Ausgabe

Der gesamte Text dieses Abschnitts wird fett gedruckt. Um den Fettdruck auszuschalten, muss die Anweisung .BF nochmals eingegeben werden, diesmal jedoch ohne Parameter.

In diesem Satz wird ein **einziges** Wort fett gedruckt. Dieses Wort muss direkt nach der Anweisung .BP eingegeben werden, und es muss in einer **extra Zeile** stehen.

Wenn Sie Umschaltanweisungen benutzen möchten, müssen Sie zuerst das Umschaltzeichen definieren. In diesem Beispiel

ist der Schrägstrich das Umschaltzeichen. Sie können nun **mehrere Wörter** oder einen Teil eines Wortes im **Fett**druck ausgeben.

Wenn Sie den Schrägstrich (/) im Text ausgeben möchten, müssen Sie das Umschaltzeichen aufheben oder ein anderes definieren.

Text unterstreichen

Zum Unterstreichen von Text gibt es zwei Anweisungen: `.UL` und `.US`. Außer diesen Anweisungen können Sie auch Umschaltanweisungen zum Unterstreichen benutzen.

Die Anweisung `.UL` muss *nach* dem zu unterstreichenden Text stehen, `.US` dagegen *vor* dem zu unterstreichenden Text.

.UL - Unterstreichen

```
.UL zeichen  
.UL
```

Die Anweisung `.UL` unterstreicht die vorhergehende Zeile des Rohtextes. Wenn Sie kein Unterstreichungszeichen angeben, wird der Bindestrich (-) benutzt. Die Unterstreichung belegt eine Ausgabezeile.

Die Anweisung `.UL` bewirkt einen Zeilenumbruch. Sie kann deshalb nicht benutzt werden, um einzelne Wörter in einer Zeile zu unterstreichen.

Geben Sie die Anweisung *nach* dem Text an, der unterstrichen werden soll, z.B.:

```
Dieser Text wird mit Bindestrichen unterstrichen.  
.UL
```

Sie können das Unterstreichungszeichen ändern. Wenn Sie z.B. den Stern (*) benutzen möchten, geben Sie an:

```
.UL *
```

Der mit dieser Anweisung unterstrichene Text kann auch zentriert (siehe [.CE - Text zentrieren](#)), rechtsbündig angeordnet (siehe [.RA - Rechtsbündige Textausgabe](#)) und fett gedruckt (siehe [.BF - Fettdruck](#)) werden.

.US - Unterstreichen

```
.US
```

Diese Anweisung unterstreicht den Text, der in der nächsten Zeile des Rohtextes steht. Es wird mit einer durchgezogenen Linie unterstrichen. Im Gegensatz zur Anweisung `.UL` können Sie hier kein Unterstreichungszeichen definieren. Es wird auch keine extra Ausgabezeile erzeugt.

Die Anweisung `.US` bewirkt keinen Zeilenumbruch, d.h. Sie können einzelne Wörter unterstreichen. Es ist nicht möglich, ein Wort teilweise zu unterstreichen (dazu sind Umschaltanweisungen erforderlich).

Geben Sie die Anweisung *vor* dem zu unterstreichenden Text ein:

```
.US  
Diese Zeile wird unterstrichen.
```

Der mit der Anweisung `.US` unterstrichene Text kann auch zentriert (siehe [.CE - Text zentrieren](#)) und rechtsbündig angeordnet (siehe [.RA - Rechtsbündige Textausgabe](#)) werden. Eine Kombination mit Fettdruck ist nur mit Umschaltanweisungen möglich.

.OP ULB - Leerzeichen unterstreichen

```
.OP ULB=ON  
.OP ULB=OFF
```

Diese Anweisung unterstreicht in der formatierten Ausgabe alle Leerzeichen. Dies entspricht der anfänglichen Einstellung:

```
.OP ULB=ON
```

Um diese Option auszuschalten, damit die Leerzeichen nicht unterstrichen werden, geben Sie an:

```
.OP ULB=OFF
```



Vorsicht: Die Anweisung `.OP ULB=OFF` wird nur erkannt, wenn der Zeilenausgleich eingeschaltet ist (`.FI ON`).

Umschaltanweisungen zum Unterstreichen von Text

Mit Umschaltanweisungen können Sie auch einen Teil eines Wortes unterstreichen und dies mit Fettdruck kombinieren. Umschaltanweisungen bewirken keinen Zeilenumbruch.

Bevor Sie eine Umschaltanweisung benutzen können, müssen Sie das Umschaltzeichen definieren (siehe *.OP ESC - Umschaltzeichen definieren*).

Um ein Wort teilweise zu unterstreichen, müssen Sie Folgendes im Rohtext angeben:

U1	Unterstreichung einschalten.
U0	Unterstreichung ausschalten.

Im folgenden Beispiel wird ein Wort zum Teil unterstrichen:

```
.OP ESC=/  
Sie können ein Wort teilweise /U1unter/U0streichen.
```

Die Ausgabe sieht folgendermaßen aus:

```
Sie können ein Wort teilweise unterstreichen.
```

Sie können die Umschaltanweisungen für Unterstreichung und Fettdruck kombinieren:

```
.OP ESC=/  
Sie können Unterstreichung und Fettdruck /M1/U1kombinieren/U0/M0.
```

Die Ausgabe sieht folgendermaßen aus:

```
Sie können Unterstreichung und Fettdruck kombinieren.
```

Beispiel

Dieses Beispiel zeigt, wie Text mit den Anweisungen `.UL` und `.US` und mit Umschaltanweisungen unterstrichen wird.

Rohtext

```
.LM 0;.RM 60
.FI ON;.JU ON
Diese Zeile wird mit dem Zeichen = unterstrichen.
.UL =
Die Anweisung .UL wird nach der Zeile angegeben, die unterstrichen
werden soll. Sie bewirkt einen Zeilenumbruch.
.SL 1
In diesem Satz wird ein
.US
einziges
Wort unterstrichen. Das Wort oder die Wörter, die unterstrichen werden
sollen, müssen auf die Anweisung .US direkt folgen und in einer
.US
extra Zeile
stehen.
.SL 1
Wenn Sie Umschaltanweisungen benutzen möchten, müssen Sie zuerst das
Umschaltzeichen definieren.
.OP ESC=/
In diesem Beispiel ist der Schrägstrich das Umschaltzeichen.
Sie können nun /Ulmehrere Wörter/U0 oder ein Wort /Ulteil/U0weise
unterstreichen.
.SL 1
Sie können Fettdruck und Unterstreichung /M1/U1kombinieren/M0/U0.
.** Die folgende Anweisung macht die Definition des Umschaltzeichens
.** wieder rückgängig.
.OP ESC=NULL
.SL 1
Wenn Sie den Schrägstrich (/) im Text ausgeben möchten, müssen Sie das
Umschaltzeichen aufheben oder ein anderes definieren.
```

Formatierte Ausgabe

```
Diese Zeile wird mit dem Zeichen = unterstrichen.
=====
Die Anweisung .UL wird nach der Zeile angegeben, die
unterstrichen werden soll. Sie bewirkt einen Zeilenumbruch.

In diesem Satz wird ein einziges Wort unterstrichen. Das
Wort oder die Wörter, die unterstrichen werden sollen,
müssen auf die Anweisung .US direkt folgen und in einer
extra Zeile stehen.
```

Wenn Sie Umschaltanweisungen benutzen möchten, müssen Sie zuerst das Umschaltzeichen definieren. In diesem Beispiel ist der Schrägstrich das Umschaltzeichen. Sie können nun mehrere Wörter oder ein Wort teilweise unterstreichen.

Sie können Fettdruck und Unterstreichung kombinieren.

Wenn Sie den Schrägstrich (/) im Text ausgeben möchten, müssen Sie das Umschaltzeichen aufheben oder ein anderes definieren.

Kursivdruck

Um Text kursiv zu drucken, müssen Sie Umschaltanweisungen verwenden. Bevor Sie eine Umschaltanweisung benutzen können, müssen Sie das Umschaltzeichen definieren (siehe **.OP ESC - Umschaltzeichen definieren**).

Um Text kursiv zu drucken, geben Sie im Rohtext Folgendes an:

I1	Kursivdruck einschalten.
I0	Kursivdruck ausschalten.

Beispiel:

```
.OP ESC=/
Dies ist ein Beispiel für /I1kursiv gedruckten Text/I0.
```

Die Ausgabe sieht folgendermaßen aus:

```
Dies ist ein Beispiel für kursiv gedruckten Text.
```

Kursivdruck und Unterstreichung können kombiniert werden. Beispiel:

```
.OP ESC=/
In diesem Beispiel wird /I1/U1kursiver Text/U0/I0 unterstrichen.
```

Die Ausgabe sieht folgendermaßen aus:

```
In diesem Beispiel wird kursiver Text unterstrichen.
```

Text hochstellen

Um Text hochgestellt zu drucken, müssen Sie Umschaltanweisungen verwenden. Bevor Sie eine Umschaltanweisung benutzen können, müssen Sie das Umschaltzeichen definieren (siehe **.OP ESC - Umschaltzeichen definieren**).

Um hochgestellten Text zu drucken, geben Sie im Rohtext Folgendes an:

E1	Hochstellen einschalten.
E0	Hochstellen ausschalten.

Beispiel:

```
.OP ESC=/  
Dies ist ein Beispiel für /E1hoch/E0gestellten Text.
```

Die Ausgabe sieht folgendermaßen aus:

```
Dies ist ein Beispiel für hochgestellten Text.
```

Hochstellen kann mit Fettdruck, Kursivdruck und Unterstreichung beliebig kombiniert werden.
Beispiel:

```
.OP ESC=/  
Hier wird /M1/E1hoch/E0/M0gestellter Text im Fettdruck ausgegeben.
```

Die Ausgabe sieht folgendermaßen aus:

```
Hier wird hochgestellter Text im Fettdruck ausgegeben.
```

Text tiefstellen

Um Text tiefgestellt zu drucken, müssen Sie Umschaltanweisungen verwenden. Bevor Sie eine Umschaltanweisung benutzen können, müssen Sie das Umschaltzeichen definieren (siehe **.OP ESC - Umschaltzeichen definieren**).

Um tiefgestellten Text zu drucken, geben Sie im Rohtext Folgendes an:

S1	Tiefstellen einschalten.
S0	Tiefstellen ausschalten.

Beispiel:

```
.OP ESC=/
Dies ist ein Beispiel für /S1tief/S0gestellten Text.
```

Die Ausgabe sieht folgendermaßen aus:

```
Dies ist ein Beispiel für tiefgestellten Text.
```

Tiefstellen kann mit Fettdruck und Kursivdruck kombiniert werden. Beispiel:

```
.OP ESC=/
Hier wird /I1/S1tief/S0/I0gestellter Text kursiv gedruckt.
```

Die Ausgabe sieht folgendermaßen aus:

```
Hier wird tiefgestellter Text kursiv gedruckt.
```

Text überdrucken

Um Text zu überdrucken, müssen Sie Umschaltanweisungen verwenden. Bevor Sie eine Umschaltanweisung benutzen können, müssen Sie das Umschaltzeichen definieren (siehe [.OP ESC - Umschaltzeichen definieren](#)).

Das Überdrucken von Text wird verwendet, um zwei Zeichen an der gleichen Position zu drucken (z.B. den dänischen Buchstaben ø).

Um Text zu überdrucken, geben Sie im Rohtext Folgendes an:

```
B Ein Zeichen zurückgehen.
```

Das folgende Beispiel zeigt, wie Sie zwei Zeichen übereinander drucken, um den dänischen Buchstaben ø auszugeben. Da der Schrägstrich (/) im Rohtext vorkommt, dürfen Sie ihn nicht als Umschaltzeichen definieren:

```
.OP ESC=#  
Dies ist ein dänisches Wort: o#/re.
```

Die Ausgabe sieht folgendermaßen aus:

```
Dies ist ein dänisches Wort: øre.
```


Umschaltanweisungen

Für folgende Text hervorhebungen ist es sinnvoll, Umschaltanweisungen zu benutzen: Kursivdruck, Hochstellen, Tiefstellen und Überdrucken. Zum Kombinieren von Hervorhebungen, z.B. Fettdruck mit Unterstreichung, müssen Umschaltanweisungen benutzt werden, da dies mit Con-form-Anweisungen nicht möglich ist.

Mit Hilfe von Umschaltanweisungen können Sie auch definieren, an welchen Stellen Wörter getrennt werden sollen (siehe [Trennstellen definieren](#)).

Der folgenden Tabelle können Sie die Symbole für die verschiedenen Text hervorhebungen entnehmen:

B	Druckposition ein Zeichen zurück zum Überdrucken von Text.
E1	Hochstellen einschalten.
E0	Hochstellen ausschalten.
I1	Kursivdruck einschalten.
I0	Kursivdruck ausschalten.
M1	Fettdruck einschalten.
M0	Fettdruck ausschalten.
U1	Unterstreichung einschalten.
U0	Unterstreichung ausschalten.
S1	Tiefstellen einschalten.
S0	Tiefstellen ausschalten.

Für jedes Symbol müssen Sie die entsprechende Druckerbefehlsfolge definieren (wie im Druckerhandbuch beschrieben). Siehe *Con-nect Benutzerhandbuch*, Abschnitt *Druckerprofile*.

Weitere Symbole für Umschaltanweisungen sind (siehe [Die Textrichtung mit Umschaltanweisungen ändern](#)):

A1	Textrichtung von-links-nach-rechts einschalten.
A0	Textrichtung von-links-nach-rechts ausschalten.
Z1	Textrichtung von-rechts-nach-links einschalten.
Z0	Textrichtung von-rechts-nach-links ausschalten.

.OP ESC - Umschaltzeichen definieren

```
.OP ESC=zeichen  
.OP ESC=NULL
```

Bevor Sie eine Umschaltanweisung eingeben können, müssen Sie das Umschaltzeichen definieren. Um z.B. den Schrägstrich (/) als Umschaltzeichen zu definieren, geben Sie an:

```
.OP ESC=/'
```

Nach dieser Anweisung können Sie den Schrägstrich nicht mehr als Text verwenden, da Con-nect ihn als Umschaltzeichen interpretiert, dem eine Umschaltanweisung folgt.

Die Definition des Umschaltzeichens aufheben

Wenn der Schrägstrich (der mit der obigen Anweisung als Umschaltzeichen definiert wurde) als Text in der formatierten Ausgabe erscheinen soll, müssen Sie entweder ein anderes Umschaltzeichen definieren oder die Definition des Umschaltzeichens mit der folgenden Anweisung rückgängig machen:

```
.OP ESC=NULL
```

Eigene Symbole für Umschaltanweisungen definieren

Sie können eigene Symbole für Umschaltanweisungen definieren. Diese Symbole müssen Sie auch in Ihrem Druckerprofil angeben (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Druckerprofile*).

Die selbstdefinierten Symbole müssen entweder in Klammern oder in Anführungszeichen eingegeben werden.

Beispiel: Sie haben das Symbol "Courier" definiert, das in den gleichnamigen Schrifttyp umschaltet. Wenn Sie Klammern verwenden, geben Sie im Rohtext Folgendes ein:

```
.OP ESC=/  
/(COURIER)
```

Wenn Sie Anführungszeichen verwenden, geben Sie im Rohtext Folgendes ein:

```
.OP ESC=/  
/"COURIER"
```

Beispiel

Dieses Beispiel zeigt, wie Umschaltanweisungen verwendet werden.

Rohtext

```
.LM 0;.RM 60
.FI ON;.JU ON
.OP ESC=/
```

Mit Umschaltanweisungen können Sie ein Wort teilweise hervorheben oder einen längeren Textabschnitt hervorheben, z.B. durch /M1Fett/M0druck, /U1Unterstreichen/U0, /I1Kursivdruck/I0, /S1Tief/S0stellen oder /E1Hoch/E0stellen. Sie können /M1/U1Fettdruck mit Unterstreichen/M0/U0 und /I1/U1Kursivdruck mit Unterstreichen/I0/U0 kombinieren.

```
.SL 1
/I1Sie können längere Textabschnitte kursiv drucken und dies mit
/S1tiefgestelltem/S0 oder /E1hochgestelltem/E0 Text kombinieren./I0
.SL 1
/M1Sie können längere Textabschnitte fett drucken und dies mit
/S1tiefgestelltem/S0 oder /E1hochgestelltem/E0 Text kombinieren./M0
.SL 1
.OP ESC=#
```

Es ist möglich, zwei Zeichen übereinander zu drucken, z.B. um das dänische Wort "o#B/re" auszugeben. Wenn Sie den Schrägstrich ausgeben möchten, müssen Sie sicherstellen, dass er nicht als das aktuelle Umschaltzeichen definiert ist.

```
.SL 1
.** Das Umschaltzeichen ist immer noch das Zeichen #.
.CE ON
Sie können Text #U1zentriert#U0 ausgeben.
.CE OFF;.SL 1
.RA ON
Sie können Text #M1rechts#M0bündig ausgeben.
.RA OFF
```

Formatierte Ausgabe

Mit Umschaltanweisungen können Sie ein Wort teilweise hervorheben oder einen längeren Textabschnitt hervorheben, z.B. durch **Fettdruck**, Unterstreichen, *Kursivdruck*, _{Tief}stellen oder ^{Hoch}stellen. Sie können **Fettdruck mit Unterstreichen** und *Kursivdruck mit Unterstreichen* kombinieren.

Sie können längere Textabschnitte kursiv drucken und dies mit _{tief}gestelltem oder ^{hoch}gestelltem Text kombinieren.

Sie können längere Textabschnitte fett drucken und dies mit

tiefgestelltem oder ^{hoch}gestelltem Text kombinieren.

Es ist möglich, zwei Zeichen übereinander zu drucken, z.B. um das dänische Wort "øre" auszugeben. Wenn Sie den Schrägstrich ausgeben möchten, müssen Sie sicherstellen, dass er nicht als das aktuelle Umschaltzeichen definiert ist.

Sie können Text zentriert ausgeben.

Sie können Text **rechtsbündig** ausgeben.

6 Silbentrennung

- Silbentrennung in der formatierten Ausgabe 141
- Im Rohtext getrennte Wörter 147

Dieses Kapitel behandelt die folgenden Themen:

Beim Blocksatz (.FI ON;JU ON) bewirkt die Silbentrennung, dass für die Ausrichtung des Textes weniger Leerzeichen zwischen den Wörtern eingefügt werden müssen.

Beim Flatterrand (.FI ON;JU OFF) bewirkt die Silbentrennung, dass der rechte Rand gleichmäßiger wird.

Die Silbentrennung ist für folgende Sprachen verfügbar: Deutsch, Englisch und Französisch.

Silbentrennung in der formatierten Ausgabe

Die folgenden Optionen betreffen nur die Wörter am Ende einer Ausgabezeile. Die Silbentrennung funktioniert nur, wenn der Zeilenausgleich eingeschaltet ist (.FI ON).

.OP HYP - Silbentrennung

```
.OP HYP=sprachcode
.OP HYP=OFF
```

Anfänglich ist die Silbentrennung ausgeschaltet. Um sie einzuschalten, müssen Sie den Sprachcode für die gewünschte Sprache definieren. Die möglichen Sprachcodes sind:

E	Englisch
F	Französisch
G	Deutsch (alte Rechtschreibung)
D	Deutsch (neue Rechtschreibung)

Um die Silbentrennung für die deutsche Sprache einzuschalten, geben Sie an:

```
.OP HYP=D
```

Solange die Silbentrennung eingeschaltet ist, werden lange Wörter am Ende einer Ausgabezeile den Silbentrennungsregeln entsprechend getrennt.

Um die Silbentrennung auszuschalten, geben Sie an:

```
.OP HYP=OFF
```

.OP HYB - Zeichen vor Trennung

`.OP HYB=zah1`

Sie können die Mindestanzahl der Zeichen festlegen, die bei einer Silbentrennung vor dem Trennstrich stehen sollen. Wenn es mindestens 2 Zeichen sein sollen, geben Sie an:

`.OP HYB=2`

In diesem Fall hat z.B. das Wort "Wochentage" folgende Trennstellen:

Wo-chen-ta-ge

Wenn Sie mindestens 3 Zeichen vor dem Trennstrich definieren, wird das Wort "Wochentage" folgendermaßen getrennt:

Wochen-tage

.OP HYA - Zeichen nach Trennung

`.OP HYA=zah1`

Sie können die Mindestanzahl der Zeichen festlegen, die bei einer Silbentrennung nach dem Trennstrich stehen sollen. Wenn es mindestens 2 Zeichen sein sollen, geben Sie an:

`.OP HYA=2`

In diesem Fall hat z.B. das Wort "Informationsaustausch" folgende Trennstellen:

In-for-ma-ti-ons-aus-tausch

Wenn Sie mindestens 4 Zeichen nach dem Trennstrich definieren, wird das Wort "Informationsaustausch" folgendermaßen getrennt:

In-forma-tions-austausch

Wenn Sie die Kombination

.OP HYA=4, HYB=4

angeben, kann das Wort nur noch an folgenden Stellen getrennt werden:

Infor-mations-austausch

Trennstellen definieren

Um zu vermeiden, dass ein Wort falsch getrennt wird, können Sie angeben, wo getrennt werden soll. Bei automatischer Trennung wird z.B. das Wort "Zeilenende" folgendermaßen getrennt:

```
Ze-i-le-nen-de
```

Um dies zu korrigieren, können Sie die richtige Trennstelle definieren:

```
Ze-i-len-en-de
```

Um Trennstellen zu definieren, müssen Sie zuerst das Umschaltzeichen bestimmen (siehe [.OP ESC - Umschaltzeichen definieren](#)). Dann können Sie die Umschaltanweisung, d.h. das Umschaltzeichen gefolgt von einem Trennstrich, eingeben:

```
.OP ESC=/  
Ein langes Wort wie Informations/-austausch muss am Zeilen/-ende getrennt werden.
```

Wenn Sie Trennstellen definieren, werden die Wörter nur noch an diesen Stellen und nicht mehr an den vordefinierten Stellen getrennt.

Die Silbentrennung für ein Wort ausschalten

Wenn Sie möchten, dass ein Wort überhaupt nicht getrennt wird, können Sie die Umschaltanweisung (d.h. das Umschaltzeichen gefolgt von einem Trennstrich) direkt vor dem betreffenden Wort angeben. Beispiel:

```
.OP ESC=/  
/-Wiese
```

Sie können die Umschaltanweisung auch nach dem Wort angeben. Beispiel:

```
.OP ESC=  
Wiese/-
```

Beispiel

Dieses Beispiel zeigt, wie Sie die Silbentrennung für die deutsche Sprache einschalten, die Mindestanzahl von Zeichen vor und nach dem Trennstrich definieren und wie Sie Trennstellen definieren.

Rohtext

```
.LM 0;.RM 50
.FI ON;.JU ON
.OP HYP=D;.** Silbentrennung für die deutsche Sprache
.OP HYB=2;.** mindestens 2 Zeichen vor dem Trennstrich
.OP HYA=3;.** mindestens 3 Zeichen nach dem Trennstrich
Sie können die Silbentrennung für folgende Sprachen angeben:
Englisch, Französisch und Deutsch. Sie können auch die
Mindestanzahl der Zeichen angeben, die vor und nach dem
Trennstrich stehen sollen.
.IL 1
Um die Trennstellen zu definieren, müssen Sie Umschaltanweisungen
benutzen. Dadurch stellen Sie sicher, dass ein Wort an der von
Ihnen angegebenen Position getrennt wird.
Sie müssen zuerst das Umschaltzeichen definieren.
.OP ESC=/
In diesem Beispiel ist es der Schrägstrich. Im Text können Sie
nun den Schrägstrich gefolgt vom Trennstrich angeben, um die
Trennstelle zu definieren.
.IL 1;.RM 42
Beispiel: Mit der Umschaltanweisung stellen Sie sicher, dass
das Wort "Positions/-angabe" richtig getrennt wird.
.OP HYP=OFF;.IL 1
Die Silbentrennung ist anfänglich ausgeschaltet.
```

Formatierte Ausgabe

Sie können die Silbentrennung für folgende Sprachen angeben: Englisch, Französisch und Deutsch. Sie können auch die Mindestanzahl der Zeichen angeben, die vor und nach dem Trennstrich stehen sollen.

Um die Trennstellen zu definieren, müssen Sie Umschaltanweisungen benutzen. Dadurch stellen Sie sicher, dass ein Wort an der von Ihnen angegebenen Position getrennt wird. Sie müssen zuerst das Umschaltzeichen definieren. In diesem Beispiel ist es der Schrägstrich. Im Text können Sie nun den Schrägstrich gefolgt vom Trennstrich angeben, um die Trennstelle zu definieren.

Beispiel: Mit der Umschaltanweisung stel-

Silbentrennung

len Sie sicher, dass das Wort "Positions-
angabe" richtig getrennt wird.

Die Silbentrennung ist anfänglich
ausgeschaltet.

Im Rohtext getrennte Wörter

Die folgende Anweisung gilt nur für Wörter, die im Rohtext am Zeilenende getrennt wurden. Sie funktioniert nur, wenn der Zeilenausgleich eingeschaltet ist (.FI ON).

.HP - Trennung aufheben

```
.HP ON  
.HP OFF
```

Sie können ein Wort am Ende einer Rohtextzeile trennen und angeben, dass die Trennung in der formatierten Ausgabe nicht vorgenommen werden soll. Geben Sie dazu die folgende Anweisung an:

```
.HP ON
```

Diese Funktion ist anfänglich ausgeschaltet, d.h. ein Wort, das am Ende einer Rohtextzeile getrennt wurde, wird auch in der formatierten Ausgabe getrennt, selbst wenn es in der Mitte einer Zeile steht. Dies entspricht der folgenden Anweisung:

```
.HP OFF
```

Beispiel

Dieses Beispiel zeigt, wie die Trennung eines Wortes am Ende einer Rohtextzeile in der formatierten Ausgabe aufgehoben werden kann.

Rohtext

```
.LM 0;.RM 60  
.FI ON;.JU ON  
.HP ON
```

Sie können angeben, dass lange Wörter, die Sie in Ihrem Rohtext am Zeilenende getrennt haben, im formatierten Dokument ungetrennt ausgegeben werden sollen. Dies ist aber nur möglich, wenn der Zeilenausgleich eingeschaltet ist. Diese Funktion gilt nicht für Wörter, die im Rohtext innerhalb einer Zeile getrennt wurden.

```
.HP OFF;.IL 1
```

Diese Funktion ist anfänglich ausgeschaltet.

Das bedeutet, dass die Wörter, die im Rohtext am Zeilenende getrennt wurden, auch in der formatierten Ausgabe getrennt werden. Außerdem wird ein Leerzeichen nach dem Trennstrich eingefügt (bei eingeschaltetem Randausgleich können es mehrere Leerzeichen sein). Dies gilt nicht für Wörter, die innerhalb einer Rohtextzeile getrennt wurden.

Formatierte Ausgabe

Sie können angeben, dass lange Wörter, die Sie in Ihrem Rohtext am Zeilenende getrennt haben, im formatierten Dokument ungetrennt ausgegeben werden sollen. Dies ist aber nur möglich, wenn der Zeilenausgleich eingeschaltet ist. Diese Funktion gilt nicht für Wörter, die im Rohtext innerhalb einer Zeile getrennt wurden.

Diese Funktion ist anfänglich ausgeschaltet. Das bedeutet, dass die Wörter, die im Rohtext am Zeilenende getrennt wurden, auch in der formatierten Ausgabe getrennt werden. Außerdem wird ein Leerzeichen nach dem Trennstrich eingefügt (bei eingeschaltetem Randausgleich können es mehrere Leerzeichen sein). Dies gilt nicht für Wörter, die innerhalb einer Rohtextzeile getrennt wurden.

7 Tabulatoren und Rahmen

- Tabulatoren 150
- Rahmen 160

Dieses Kapitel behandelt die folgenden Themen:

Tabulatoren

Wenn Sie mit Hilfe von Tabulatoren eine Tabelle erstellen, wird empfohlen, den Zeilenausgleich auszuschalten (.FI OFF), damit die Spalten der Tabelle nicht verschoben werden. Wenn der Zeilenausgleich eingeschaltet ist, setzt Con-form zuerst die Tabulatoren und führt dann die Anweisung .FI ON aus. Dies kann dazu führen, dass sich die Spalten verschieben.

Das Tabulatorzeichen ist das Zeichen, das Sie im Rohtext eingeben, um den darauf folgenden Text an die definierte Tabulatorposition zu verschieben. Außer dem Leerzeichen können Sie jedes beliebige Zeichen als Tabulatorzeichen angeben. Wenn Sie das Anweisungstrennzeichen (anfänglich das Semikolon) als Tabulatorzeichen definieren möchten, müssen Sie das Zeichen doppelt eingeben (;).



Anmerkung: Jeder Tabulator belegt mindestens eine Leerstelle.

Die Tabulatorposition definiert die Spalte, in der der Tabulator sitzen soll. Diese Position wird vom aktuellen linken Rand beeinflusst, der mit der Anweisung `.LM` definiert wurde. Wenn Sie z.B. für den linken Rand `.LM 10` angeben, beginnt der Text in Spalte 11. Definieren Sie dann einen Tabulator 10 Spalten rechts vom linken Rand, beginnt der dem Tabulator folgende Text in Spalte 21.



Anmerkung: Wenn der linke Rand mit der Anweisung `.TI` oder `.OF` verschoben wird, verschieben sich alle Tabulatorpositionen entsprechend.

Um mehrere Tabulatoren zu definieren, müssen Sie die Spaltenangaben durch Kommas voneinander trennen; dabei dürfen keine Leerzeichen angegeben werden:

```
.TB *=10,20,30,40
```

Vor den Spaltennummern - außer vor der ersten - kann ein Pluszeichen (+) stehen. Dies bewirkt, dass der Tabulator n Spalten rechts vom vorherigen Tabulator gesetzt wird. Die beiden folgenden Anweisungen sind daher in ihrem Ergebnis identisch:

```
.TB *=5,10,18,23,30,40,50,65
```

```
.TB *=5,+5,+8,+5,+7,+10,+10,+15
```

Sie können höchstens 20 Tabulatoren definieren.

Mit jeder neuen `.TB`-Anweisung werden die vorher definierten Tabulatoren gelöscht. Die Tabulatortaste Ihrer Tastatur wird von Con-form nicht benutzt.

Tabulatoren ausrichten

Um einen Tabulator auszurichten, geben Sie nach der Spaltennummer einen der folgenden Buchstaben an:

L	linksbündig anordnen
C	zentrieren
R	rechtsbündig anordnen
D	Dezimaltabulator

Linksbündiger Tabulator

Um einen linksbündigen Tabulator zu setzen, geben Sie den Buchstaben L nach der Spaltennummer an, z.B.:

```
.TB *=10L
```

Der Text wird so ausgerichtet, dass er in der angegebenen Spalte beginnt.



Anmerkung: Wenn Sie keinen Buchstaben angeben, werden alle Tabulatoren linksbündig ausgerichtet.

Zentrierter Tabulator

Um einen zentrierten Tabulator zu setzen, geben Sie den Buchstaben C nach der Spaltennummer an, z.B.:

```
.TB *=30C
```

Der Text wird so ausgerichtet, dass seine Mitte in der angegebenen Spalte liegt.

Rechtsbündiger Tabulator

Um einen rechtsbündigen Tabulator zu setzen, geben Sie den Buchstaben R nach der Spaltennummer an, z.B.:

```
.TB *=50R
```

Der Text wird so ausgerichtet, dass er in der angegebenen Spalte endet.

Dezimaltabulator

Um einen Dezimaltabulator zu setzen, geben Sie den Buchstaben D nach der Spaltennummer an, z.B.:

```
.TB *=25D
```

Der Text wird so ausgerichtet, dass das Dezimalzeichen in der angegebenen Spalte steht. Wenn der Text kein Dezimalzeichen enthält, wird angenommen, es stünde direkt rechts neben dem Text.

Anfänglich ist der Punkt (.) das Dezimalzeichen. Sie können das Dezimalzeichen ändern. Um z.B. das Komma als Dezimalzeichen zu definieren, geben Sie an:

```
.OP DEC=' , '
```

Da das Komma bei bestimmten Con-form-Anweisungen als Parametertrennzeichen benutzt wird, ist es wichtig, das Komma (wie in der obigen Anweisung) in Anführungszeichen zu setzen. Um z.B. den Schrägstrich als Dezimalzeichen zu definieren, müssen Sie keine Anführungszeichen angeben (.OP DEC=).

Füllzeichen

Mit jedem Tabulator werden Füllzeichen ausgegeben. Sie können mit Ausnahme des Gleichheitszeichens (=) jedes beliebige Zeichen als Füllzeichen definieren. Wenn Sie das Anweisungstrennzeichen (anfänglich das Semikolon) als Füllzeichen angeben möchten, müssen Sie es doppelt eingeben (;:). Wenn Sie kein Füllzeichen definieren, werden Leerzeichen benutzt.

Wenn Sie den Punkt (.) als Füllzeichen definieren möchten, geben Sie ihn vor dem Gleichheitszeichen ein:

```
.TB *. =10
```

Beispiel:

```
.FI OFF
.TB *. =15
Athen*07:48
Frankfurt*09:05
London*10:34
```

Dieser Rohtext ergibt die folgende formatierte Ausgabe:

```
Athen.....07:48
Frankfurt.....09:05
London.....10:34
```

Wenn Sie einen der Buchstaben verwenden, die für die Ausrichtung des Tabulators definiert sind (L, C, R oder D), können Sie das Füllzeichen auch nach dem Gleichheitszeichen angeben. In diesem Fall müssen Sie das Füllzeichen zwischen der Spaltennummer und dem Buchstaben eingeben. Sie können dann für jeden Tabulator ein anderes Füllzeichen angeben, z.B.:

```
.TB *=10-L,30.C,50_R
```

Die Füllzeichen werden immer *vor* dem definierten Tabulator ausgegeben.



Anmerkung: Wenn Sie vor und nach dem Gleichheitszeichen Füllzeichen angegeben haben, wird das Zeichen vor dem Gleichheitszeichen nur dann berücksichtigt, wenn keine Spaltennummer (nach dem Gleichheitszeichen) vor dem Füllzeichen steht.

Die Definition des Tabulatorzeichens rückgängig machen

Wenn Sie das Zeichen, das als Tabulatorzeichen definiert ist, ausgeben möchten, können Sie die Definition des Tabulatorzeichens mit der folgenden Anweisung rückgängig machen:

```
.TB OFF
```

Sie können das Tabulatorzeichen auch ändern. Wenn das aktuelle Tabulatorzeichen der Stern ist und Sie stattdessen das Nummernzeichen (#) benutzen möchten, geben Sie an:

```
.TB #=10
```

Beispiel

Dieses Beispiel besteht aus drei Teilen. Der erste Teil zeigt, wie mit Hilfe von Tabulatoren eine Tabelle erstellt wird, die am Dezimalpunkt ausgerichtet wird. Der zweite Teil zeigt, dass das Tabulatorzeichen von der Anweisung **.IC** nicht ignoriert wird. Der dritte Teil zeigt, wie Tabulatoren mit der Anweisung **.OF** benutzt werden. Die Anweisung **.OF** definiert den linken Rand für alle Zeilen außer der ersten Zeile.

Rohtext

```
.LM 0;.RM 60
.FI OFF
.SL 1
.** Das folgende Beispiel zeigt, wie eine einfache Tabelle angelegt
.** wird. Die beiden letzten Spalten werden mit Dezimaltabulatoren
.** ausgerichtet.
.TB *=10,30D,40D
2 Dutzend*Eier*3.72*7.44
3 Liter*Milch*.99*2.97
1 Pfund*Butter*2.29*2.29
2 Kilo*Fleisch*15.99*31.98
.SL 1
Summe***44.68
.SL 2
.** Das folgende Beispiel zeigt, dass das Tabulatorzeichen von der
.** Anweisung .IC nicht ignoriert wird.
Die folgenden Anweisungen bewirken einen Zeilenumbruch:
.TB #=10,20,30,40
.IC ON
.BF#.EF#.IP#.OF#.SL
.BX#.EM#.LM#.PH#.TI
.CE#.FI#.NL#.PS#.UL
.CH#.IL#.NP#.RA
.IC OFF
```

```
.SL 2
.** Das folgende Beispiel zeigt, wie Tabulatoren mit .OF benutzt werden.
.FI ON;.JU ON
.TB *=15
.OF 15
Einzug:*Der Tabulator wurde so gesetzt, dass der restliche Text
der ersten Zeile in der gleichen Spalte beginnt wie der Text,
der eingerückt wurde.
```

Formatierte Ausgabe

2 Dutzend Eier		3.72	7.44
3 Liter Milch		.99	2.97
1 Pfund Butter		2.29	2.29
2 Kilo Fleisch		15.99	31.98
Summe			44.68

Die folgenden Anweisungen bewirken einen Zeilenumbruch:

.BF	.EF	.IP	.OF	.SL
.BX	.EM	.LM	.PH	.TI
.CE	.FI	.NL	.PS	.UL
.CH	.IL	.NP	.RA	

Einzug: Der Tabulator wurde so gesetzt, dass der restliche Text der ersten Zeile in der gleichen Spalte beginnt wie der Text, der eingerückt wurde.

Rahmen

Sie können leere Rahmen erstellen oder Rahmen, in denen Text steht. Mit Hilfe von Rahmen können Sie auch Tabellen anlegen.

.BX - Rahmen zeichnen

```
.BX [-]zahl [, [+]zahl]... , [±]zahl
.BX
.BX OFF
.BX CANCEL
```

Um einen Rahmen zu zeichnen, müssen Sie folgendermaßen vorgehen:

1. Definieren Sie die Positionen der senkrechten Linien. Wenn der Rahmen z.B. in Spalte 10 anfangen und in Spalte 55 enden soll, geben Sie an:

```
.BX 10,55
```

Diese Anweisung zeichnet die obere waagerechte Linie des Rahmens. Die angegebenen Spaltennummern bestimmen, wo die waagerechte Linie beginnt und endet. Sie bestimmen auch die Positionen der senkrechten Linien des Rahmens.



Anmerkung: Wenn der erste Parameter 0 ist, wird die linke senkrechte Linie nicht gezeichnet.

Sie müssen bei der Anweisung .BX mindestens zwei Parameter angeben. Maximal sind 20 Parameter möglich.

Vor jeder Spaltennummer außer der ersten kann ein Pluszeichen (+) stehen. Damit geben Sie an, dass die senkrechte Linie n Spalten rechts neben der vorhergehenden Linie gezeichnet werden soll. Die beiden folgenden Anweisungen sind daher in ihrem Ergebnis identisch:

```
.BX 10,55
```

```
.BX 10,+45
```

Die mit der Anweisung .BX angegebenen Spaltennummern sind (anders als bei der Anweisung .TB) absolute Positionen. Sie werden vom aktuellen linken Rand nicht beeinflusst.

2. Geben Sie die Höhe des Rahmens an. Die Anzahl der Zeilen innerhalb des Rahmens ergibt sich aus dem Text, den Sie nun eingeben und/oder der Anzahl von Leerzeilen, die Sie angeben.

```
.IL 5
```

Diese Anweisung bewirkt, dass die beiden senkrechten Linien an den angegebenen Positionen gezeichnet werden. Zwischen den senkrechten Linien steht kein Text.

- Um die untere waagerechte Linie zu zeichnen und wieder auf normale Textverarbeitung umzuschalten, geben Sie an:

```
.BX OFF
```

Die Anweisungen der Schritte 1 bis 3 ergeben den folgenden Rahmen:

```
+-----+
|       |
|       |
|       |
|       |
|       |
+-----+
```



Anmerkung: Einige Geräte benutzen das Ausrufezeichen (!) anstelle des senkrechten Striches. Das Ausrufezeichen wurde auch hier in den Beispielen verwendet.

Waagerechte Linien im Rahmen

Um mehrere waagerechte Linien innerhalb des Rahmens zu zeichnen, müssen Sie die Anweisung `.BX` ohne Parameter angeben. Beispiel:

```
.BX 10,55
.IL 1
.BX
.IL 1
.BX
.IL 1
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:



Ein Gitter zeichnen

Um ein Gitter mit mehr als zwei senkrechten Linien zu zeichnen, müssen Sie bei der ersten `.BX`-Anweisung mehr als zwei Parameter angeben. Die nachfolgenden `.BX`-Anweisungen (die keine Parameter haben) zeichnen Linien an den angegebenen Positionen. Beispiel:

```
.BX 10,15,20,25,30,35,40
.IL 1
.BX
.IL 1
.BX
.IL 1
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

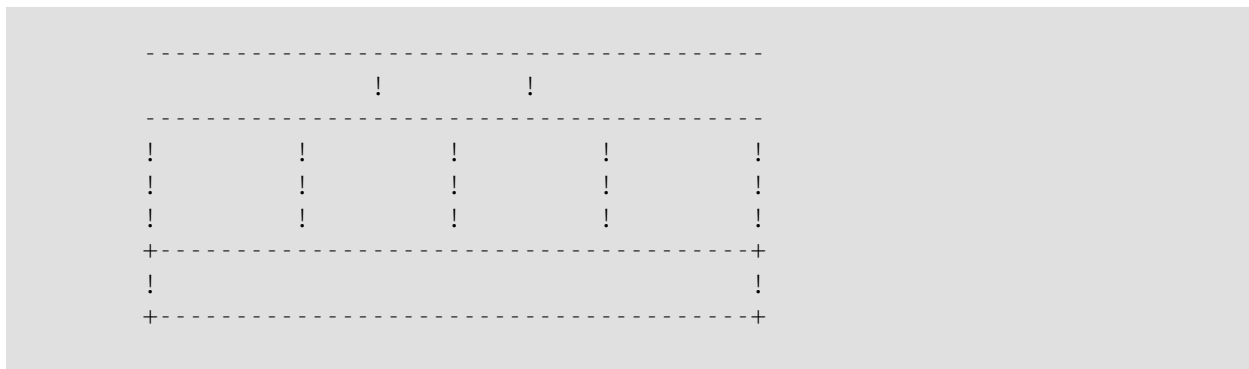
```
+-----+
!     !     !     !     !     !     !
+-----+
!     !     !     !     !     !     !
+-----+
!     !     !     !     !     !     !
+-----+
```

Übereinander liegende Rahmen

Um mehrere Rahmen übereinander zu zeichnen, deren senkrechte Linien verschiedene Positionen haben, müssen Sie die Anweisung `.BX` mit den neuen Parametern wiederholen. Wenn für einen Parameter das Minuszeichen eingegeben wird, erscheint die senkrechte Linie in dieser Spalte nicht in der Ausgabe (dies wird später noch ausführlicher erklärt). Beispiel:

```
.BX -10,25,35,-50
.IL 1
.BX 10,20,30,40,50
.IL 3
.BX 10,50
.IL 1
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:



Rahmen, die Text enthalten

Wenn Ihr Text innerhalb des Rahmens liegen soll, müssen Sie sicherstellen, dass der Parameter für den rechten Rand (Anweisung `.RM`) kleiner ist als der Parameter für die rechte senkrechte Linie. Die Anzahl der Linien innerhalb des Rahmens wird bestimmt durch den Text, den Sie nach der Anweisung `.BX` eingeben. Beispiel:

```
.FI ON;.JU ON
.BX 10,55
.LM 10;.RM 54
```

Der Parameter für den rechten Rand muss kleiner sein als der Parameter für die rechte senkrechte Linie. Andernfalls wird der rechte Rand des Rahmens nicht korrekt ausgegeben.

```
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
+-----+
!Der Parameter für den rechten Rand muss!
!kleiner sein als der Parameter für die!
!rechte senkrechte Linie. Andernfalls wird!
!der rechte Rand des Rahmens nicht korrekt!
!ausgegeben.                               !
+-----+
```

Wenn Sie den Text mit Flatterrand ausgeben möchten, müssen Sie den Randausgleich ausschalten. Den Zeilenausgleich dürfen Sie jedoch nicht ausschalten (`.FI ON;JU OFF`).

```
+-----+
!Wenn der Zeilenausgleich ausgeschaltet ist, erscheint der
!Text so, wie Sie ihn eingegeben haben. D.h. der definierte
!rechte Rand wird nicht berücksichtigt.      !
+-----+
```

Wenn Ihr Text innerhalb des Rahmens liegen soll, müssen Sie darauf achten, dass Sie den linken und den rechten Rand für den Text entsprechend angeben.

```
+-----+
Dieses Beispiel zeigt, was! passiert,
wenn! Sie nicht darauf achten, dass der
linke und der rechte Rand richtig
eingestellt sind.          !
+-----+
```

Sie können den linken und den rechten Rand so definieren, dass vor und nach einer Textzeile Leerzeichen stehen. Vor und nach dem Text können Sie auch Leerzeilen einfügen. Beispiel:

```
.FI ON;.JU ON
.BX 10,55
.LM 12;.RM 52
.IL 1
Hier sind die Ränder des Textes so eingestellt, dass vor und nach jeder
Textzeile im Rahmen jeweils 2 Leerzeichen stehen.
Außerdem wird über und unter dem Text je eine Leerzeile ausgegeben.
.IL 1
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
+-----+
!
! Hier sind die Ränder des Textes so
! eingestellt, dass vor und nach jeder
! Textzeile im Rahmen jeweils 2
! Leerzeichen stehen. Außerdem wird über
! und unter dem Text je eine Leerzeile
! ausgegeben.
!
!
+-----+
```

Rahmen ohne linke senkrechte Linie

Um einen Rahmen ohne linke senkrechte Linie zu definieren, müssen Sie ein Minuszeichen (-) vor dem Parameter für die erste Spalte angeben. Beispiel:

```
.BX -10,50
```

Die folgenden Beispiele zeigen, wie diese Funktion eingesetzt werden kann:

```
.FI ON;.JU ON
.LM 10;.RM 50
.BX -10,51
```

Die Werte für die linke senkrechte Linie und den linken Rand sind gleich.

```
.BX OFF
```

```
.IL 1
```

```
.BX -11,51
```

Der Wert für die linke senkrechte Linie wurde um eins erhöht. Nun beginnt der Rahmen in der gleichen Spalte wie der Text.

```
.BX OFF
```

```
.IL 1
```

```
.BX -15,51
```

Hier ist der Wert für die linke senkrechte Linie größer als der Wert für den linken Rand.

```
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
-----
Die Werte für die linke senkrechte Linie!
und den linken Rand sind gleich.      !
-----
```

```
-----
Der Wert für die linke senkrechte Linie!
wurde um eins erhöht. Nun beginnt der!
Rahmen in der gleichen Spalte wie der!
Text.                                  !
-----
```

```
-----
Hier ist der Wert für die linke!
senkrechte Linie größer als der Wert für!
den linken Rand.                      !
-----
```

Rahmen ohne rechte senkrechte Linie

Um einen Rahmen ohne rechte senkrechte Linie zu definieren, müssen Sie ein Minuszeichen (-) vor dem Parameter für die letzte Spalte angeben. Beispiel:

```
.BX 10,-54
```

Die folgenden Beispiele zeigen, wie diese Funktion eingesetzt werden kann:

```
.FI ON;.JU ON
.LM 10;.RM 54
.BX 10,-54
```

Die Werte für die rechte senkrechte Linie und den rechten Rand sind gleich.

```
.BX OFF
.IL 1
.BX 10,-60
```

Der Wert für die rechte senkrechte Linie ist größer als der Wert für den rechten Rand.

```
.BX OFF
.IL 1
.BX 10,-45
```

Hier ist der Wert für die rechte senkrechte Linie kleiner als der Wert für den rechten Rand.

```
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
+-----+
!Die Werte für die rechte senkrechte Linie
!und den rechten Rand sind gleich.
+-----+
```

```
+-----+
!Der Wert für die rechte senkrechte Linie ist
!größer als der Wert für den rechten Rand.
+-----+
```

```
+-----+
!Hier ist der Wert für die rechte senkrechte
!Linie kleiner als der Wert für den rechten
!Rand.
+-----+
```

Rahmen ohne untere waagerechte Linie

Um einen Rahmen ohne unteren Rand zu definieren, geben Sie die Anweisung `.BX CANCEL` anstelle der Anweisung `.BX OFF` ein. In diesem Fall wird die untere waagerechte Linie weggelassen.
Beispiel:

```
.BX 10,55
.IL 2
.BX CANCEL
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
+-----+
!                                     !
!                                     !
```

Sie können auch Rahmen ohne unteren Rand und ohne linke oder rechte senkrechte Linie zeichnen.
Beispiel:

```
.FI ON;.JU ON
.LM 12;.RM 30
.BX 10,-33
.IL 1
Dieser Rahmen hat keinen rechten und keinen unteren Rand.
.IL 1
.BX CANCEL
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
+-----+
!
! Dieser Rahmen hat
! keinen rechten und
! keinen unteren
! Rand.
!
```



Anmerkung: Die obere waagerechte Linie eines Rahmens kann nicht weggelassen werden.

Tabellen

Um eine Tabelle anzulegen, geben Sie die Tabulatoren mit der Anweisung `.TB` an und die Linien mit der Anweisung `.BX`. Beispiel:

```
.LM 0;.RM 65
.TB #=10,22
.BX 0,9,21,-67
Option#Wert#Beschreibung
.BX
BIN#Zahl#Papierkassette auswählen
BRN#ON/OFF#Zeilenumbruch unterdrücken
BXH#Zeichen#Zeichen für waagerechte Linien definieren
BXV#Zeichen#Zeichen für senkrechte Linien definieren
CHA#NUM/UNN#Kapitelnumerierung
.BX CANCEL
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
-----
Option ! Wert      ! Beschreibung
-----
BIN     ! Zahl         ! Papierkassette auswählen
BRN     ! ON/OFF       ! Zeilenumbruch unterdrücken
BXH     ! Zeichen      ! Zeichen für waagerechte Linien definieren
BXV     ! Zeichen      ! Zeichen für senkrechte Linien definieren
CHA     ! NUM/UNN     ! Kapitelnumerierung
```

Im folgenden Beispiel wurde der Text der dritten Tabellenspalte mit der Anweisung **.OF** eingerückt. D.h. die Anweisung **.OF** setzt den linken Rand für den gesamten Text nach der ersten Zeile.

```
.LM 2;.RM 58
.FI ON;.JU OFF
.BX 1,13,23,60
.TB #=12,22
.OF +22
Kapitel 6#.OP HYP#Den Sprachcode definieren
.BX;.OF +22
#.OP HYB#Anzahl der Zeichen vor der Trennung
.BX;.OF +22
#.HP#Ein Wort, das am Ende einer Zeile im Rohrtex getrennt wurde,
wird in der formatierten Ausgabe nicht getrennt.
.BX;.OF +22
Kapitel 7#.TB#Tabulatorpositionen definieren
.BX;.OF +22
#.BX#Einen Rahmen zeichnen
.BX;.OF +22
#.OP BXH#Das Zeichen für die waagerechten Linien des Rahmens definieren
.BX;.OF +22
#.OP BXV#Das Zeichen für die senkrechten Linien des Rahmens definieren
.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
+-----+
! Kapitel 6 ! .OP HYP ! Den Sprachcode definieren      !
+-----+
!           ! .OP HYB ! Anzahl der Zeichen vor der      !
!           !           ! Trennung                          !
+-----+
!           ! .HP      ! Ein Wort, das am Ende einer Zeile !
!           !           ! im Rohrtex getrennt wurde, wird in !
!           !           ! der formatierten Ausgabe nicht    !
!           !           ! getrennt.                          !
+-----+
! Kapitel 7 ! .TB      ! Tabulatorpositionen definieren    !
+-----+
!           ! .BX      ! Einen Rahmen zeichnen             !
+-----+
!           ! .OP BXH ! Das Zeichen für die waagerechten  !
!           !           ! Linien des Rahmens definieren     !
+-----+
!           ! .OP BXV ! Das Zeichen für die senkrechten  !
!           !           ! Linien des Rahmens definieren     !
+-----+
```

.OP BXH - Waagerechte Linie für Rahmen

```
.OP BXH=zeichen
```

Das Zeichen für die waagerechten Linien eines Rahmens ist der Bindestrich (-). Sie können das Zeichen ändern. Um z.B. den Stern (*) für die waagerechten Linien zu benutzen, geben Sie an:

```
.OP BXH=*
```

Wenn Sie das Zeichen für die waagerechten Linien ändern, ist das Zeichen für die Ecken des Rahmens (+) nicht mehr wirksam.

.OP BXV - Senkrechte Linie für Rahmen

```
.OP BXV=zeichen
```

Das Zeichen für die senkrechten Linien eines Rahmens ist der senkrechte Strich (|). Sie können das Zeichen ändern. Um z.B. den Stern (*) für die senkrechten Linien zu benutzen, geben Sie an:

```
.OP BXV=*
```

 **Anmerkung:** Einige Geräte benutzen das Ausrufezeichen (!) anstelle des senkrechten Striches.

Das folgende Beispiel erstellt einen Rahmen, dessen Linien mit Sternen gezeichnet werden:

```
.OP BXH=*\n.OP BXV=*\n.BX 10,55\n.I L 2\n.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
*****\n*\n*\n*****
```


8 Große Dokumente

- Dokumente einbetten 175
- Einstellungen speichern und wiederherstellen 179
- Kapitelüberschriften und Inhaltsverzeichnis 181
- Indexeinträge und Index 202

Dieses Kapitel behandelt die folgenden Themen:

Wenn Sie Kapitelüberschriften definieren, wird automatisch ein Inhaltsverzeichnis erstellt. Wenn Sie Indexeinträge angeben, wird automatisch ein Index erstellt.

Die verschiedenen Teile eines formatierten Dokuments werden in der folgenden Reihenfolge ausgegeben:

1. Dokumenttext (einschließlich aller eingebetteten Dokumente an den angegebenen Positionen)
2. Index
3. Inhaltsverzeichnis

Dokumente einbetten

Mit der Anweisung `.EM` können Sie mehrere Rohdokumente zu einem einzigen Ausgabedokument zusammenstellen.

Die einzubettenden Dokumente können entweder kurze Texte (z.B. Standardformulierungen für Briefe) oder lange Texte (z.B. ein Kapitel eines Buches) enthalten.

Wenn Sie ein Buch schreiben, können Sie ein Dokument anlegen, das alle `.EM`-Anweisungen enthält und somit die Ausgabe aller Kapitel des Buches steuert.

.EM - Einbetten

```
.EM dokumentname
```

Um ein Dokument in das aktuelle Dokument (d.h. in das Dokument, das die Anweisung `.EM` enthält) einzubetten, müssen Sie den Namen des einzubettenden Dokuments als Parameter der Anweisung `.EM` angeben. Um z.B. das Dokument namens "Vorwort" einzubetten, geben Sie an:

```
.EM Vorwort
```

Wenn der Dokumentname aus mehreren Wörtern besteht (d.h. wenn er Leerzeichen enthält), muss er in einfache Anführungszeichen eingeschlossen werden:

```
.EM 'Neue Strategie'
```

Am Ende eines eingebetteten Dokuments ist keine spezielle Anweisung erforderlich. Con-form kehrt automatisch zu dem Dokument zurück, das die Anweisung `.EM` enthält, und fährt nach der Anweisung `.EM` mit der Verarbeitung fort. Wenn es jedoch erforderlich ist, können Sie die Anweisung `.EF` (siehe unten) verwenden.

Das in der Anweisung `.EM` angegebene Dokument kann selbst auch `.EM`-Anweisungen enthalten. Diese Schachtelung ist maximal bis zur Schachtelungstiefe 5 möglich.

.EF - Ende der Datei

.EF

Wenn Sie nur den Anfang eines Dokuments einbetten möchten, können Sie im einzubettenden Dokument die Anweisung .EF angeben. Diese Anweisung bewirkt, dass Con-form zu dem Dokument zurückkehrt, das die Anweisung .EM enthält, und mit der Verarbeitung nach der Anweisung .EM fortfährt.

Alle Informationen, die im eingebetteten Dokument nach der Anweisung .EF stehen, werden nicht berücksichtigt.

.EP - Ende der Verarbeitung

```
.EP
```

Sie können diese Anweisung in jedem Dokument benutzen, um das Formatieren zu beenden (es muss nicht unbedingt ein eingebettetes Dokument sein).

Wenn die Anweisung .EP in einem eingebetteten Dokument benutzt wird, kehrt Con-form nicht zu dem Dokument zurück, das die Anweisung .EM enthält.

Mit einer **.IF-Anweisung** ist es nicht möglich, die Verarbeitung von .EP bedingt zu unterdrücken. Das Formatieren wird immer beendet.

.OP EMN - Kein Zeilenumbruch beim Einbetten

```
.OP EMN=ON  
.OP EMN=OFF
```

Sie können die Anweisung .OP EMN=ON nur benutzen, wenn der Zeilenausgleich eingeschaltet ist (.FI ON).

Wenn Sie ein Dokument in ein anderes Dokument einbetten, beginnt der Text *nach* dem eingebetteten Dokument immer in einer neuen Zeile - auch wenn der Zeilenausgleich eingeschaltet ist.

Um den Zeilenumbruch zu verhindern, geben Sie an:

```
.OP EMN=ON
```

Daraufhin fährt der Text des aktuellen Dokuments in derselben Zeile fort, in der der Text des eingebetteten Dokuments endet.

Diese Option ist anfänglich ausgeschaltet. Dies entspricht der Anweisung:

```
.OP EMN=OFF
```

Beispiel

Dieses Beispiel zeigt, wie ein Dokument eingebettet wird. Wenn das eingebettete Dokument selbst keine Anweisungen enthält, wird es gemäß der Anweisungen formatiert, die in dem Dokument angegeben sind, das die Anweisung `.EM` enthält. Wenn das eingebettete Dokument jedoch Anweisungen (z.B. Werte für die Ränder, Zeilenausgleich oder Randausgleich) enthält, so gelten diese auch für das Dokument, das die Anweisung `.EM` enthält. Es liegt an Ihnen, diese Werte am Ende des eingebetteten Dokuments wieder zurückzusetzen.

Rohtext des Dokuments, das eingebettet wird

```
.LM 10;.RM 60
.FI ON;.JU OFF
Dieses Dokument soll in ein anderes Dokument eingebettet werden.
Dabei wird es den hier angegebenen Anweisungen entsprechend formatiert.
.LM 0;.RM 70
.FI ON;.JU ON
```

Rohtext des Dokuments, das die Anweisung `.EM` enthält

```
.LM 0;.RM 70
.FI ON;.JU ON
Um Text aus einem anderen Dokument in das aktuelle Dokument einzufügen,
müssen Sie die Anweisung .EM mit dem Namen des einzubettenden Dokuments
als Parameter angeben.
.IL 1
.EM bsp
.IL 1
Wenn Sie die im eingebetteten Dokument benutzten Werte nicht
zurücksetzen, gelten sie auch für das Hauptdokument.
```

Formatierte Ausgabe

Um Text aus einem anderen Dokument in das aktuelle Dokument einzufügen, müssen Sie die Anweisung `.EM` mit dem Namen des einzubettenden Dokuments als Parameter angeben.

```
Dieses Dokument soll in ein anderes Dokument
eingebettet werden. Dabei wird es den hier
angegebenen Anweisungen entsprechend formatiert.
```

Wenn Sie die im eingebetteten Dokument benutzten Werte nicht zurücksetzen, gelten sie auch für das Hauptdokument.

Einstellungen speichern und wiederherstellen

Wenn ein Teil Ihres Dokuments anders als das restliche Dokument formatiert werden soll, können Sie die aktuellen Einstellungen mit der Anweisung `.SA` speichern. Für den Text, der auf die Anweisung `.SA` folgt, können Sie dann andere Einstellungen definieren. Später geben Sie die Anweisung `.RS` an, um die gespeicherten Einstellungen wieder zu aktivieren.



Anmerkung: Diese Funktion ist hilfreich beim Einbetten von Dokumenten.

.SA - Kontext speichern

```
.SA
```

Mit der Anweisung `.SA` können Sie die aktuellen Einstellungen des Dokuments speichern. Anschließend können Sie neue Einstellungen definieren. Sie können z.B. für den Text, der auf die Anweisung `.SA` folgt, einen anderen linken und rechten Rand angeben.

Die folgenden Funktionen beinhalten automatisch die Anweisung `.SA`:

- `.NL KEEP`
- `.NL FLOAT`
- die Verarbeitung von Fußnoten (`.FN`)
- die Verarbeitung von Kopfzeilen (`.HL`)

Mit der Anweisung `.RS` (siehe unten) stellen Sie die Einstellungen, die Sie mit der letzten `.SA`-Anweisung gespeichert haben, wieder her.

.RS - Kontext wiederherstellen

```
.RS
```

Die Anweisung `.RS` stellt die Einstellungen wieder her, die mit der letzten `.SA`-Anweisung gespeichert wurden. D.h. die Formatierung wird gemäß der Einstellungen fortgesetzt, die vor Ausführung der letzten `.SA`-Anweisung gültig waren.

Die folgenden Funktionen beinhalten automatisch die Anweisung `.RS`:

- `.NL OFF`
- das Ende der Verarbeitung von Fußnoten (`.FN`)
- das Ende der Verarbeitung von Kopfzeilen (`.HL`)

Beispiel

Dieses Beispiel zeigt, wie die Anweisungen `.SA` und `.RS` eingesetzt werden.

Rohtext

```
.LM 0;.RM 70
.FI ON;.JU ON
Mit der Anweisung .SA können Sie die aktuellen Einstellungen speichern
(z.B. die Ränder, den Zeilenausgleich oder Randausgleich).
.SA
.** Hier werden andere Einstellungen für die folgenden Zeilen definiert.
.LM 10;.RM 50
.FI ON;.JU OFF
.IL 1
Für diesen Textteil wurden andere Ränder definiert, und der Randausgleich
wurde ausgeschaltet.
.IL 1
.** Nun werden die vorherigen Einstellungen wieder aktiviert.
.RS
Mit der Anweisung .RS werden die vorher gespeicherten Einstellungen
wieder aktiviert. Die Formatierung erfolgt gemäß der Einstellungen, die
wirksam waren, bevor die Anweisung .SA ausgeführt wurde.
```

Formatierte Ausgabe

Mit der Anweisung `.SA` können Sie die aktuellen Einstellungen speichern (z.B. die Ränder, den Zeilenausgleich oder Randausgleich).

Für diesen Textteil wurden andere Ränder definiert, und der Randausgleich wurde ausgeschaltet.

Mit der Anweisung `.RS` werden die vorher gespeicherten Einstellungen wieder aktiviert. Die Formatierung erfolgt gemäß der Einstellungen, die wirksam waren, bevor die Anweisung `.SA` ausgeführt wurde.

Kapitelüberschriften und Inhaltsverzeichnis

Mit der Anweisung `.CH` definieren Sie eine Kapitelüberschrift. Diese Überschrift wird sowohl im Text als auch im Inhaltsverzeichnis ausgegeben. Das Inhaltsverzeichnis wird automatisch am Ende des Dokuments ausgegeben, beginnend auf einer neuen Seite. Wenn Sie auch Indexeinträge angegeben haben, wird das Inhaltsverzeichnis nach dem Index ausgegeben.

Mit der Anweisung `.PT` können Sie das Layout des Inhaltsverzeichnisses beeinflussen.

.CH - Kapitel

```
.CH zahl
.CH +zahl
.CH -zahl
.CH
```

Mit der Anweisung `.CH` definieren Sie Kapitelüberschriften und Einträge für das Inhaltsverzeichnis. Der dieser Anweisung folgende Text - vor dem nächsten Zeilenumbruch - ist die Überschrift des Kapitels. Beispiel:

```
.CH 1
Erstes Kapitel
```

Die Überschrift wird an der aktuellen Textposition und im Inhaltsverzeichnis ausgegeben. Der Parameter hinter der `.CH`-Anweisung definiert die Stufe (siehe [Verschiedene Stufen für Überschriften definieren](#)).

Con-form prüft, ob auf der aktuellen Seite mindestens sieben freie Zeilen verfügbar sind (d.h. Con-form führt automatisch die Anweisung `.NL 7` aus). Sind keine sieben Zeilen verfügbar, beginnt Con-form eine neue Seite und gibt die Überschrift dort aus. Damit wird sichergestellt, dass ein Kapitel oder ein Abschnitt nicht am Seitenende anfängt.

Wichtig: Zeilenumbruch nach der Überschrift

Wenn der Zeilenausgleich eingeschaltet ist (.FI ON), muss nach der Überschrift ein Zeilenumbruch erfolgen. Wenn nicht, wird der gesamte Text zwischen der Anweisung .CH und dem nächsten Zeilenumbruch als Überschrift interpretiert und in das Inhaltsverzeichnis aufgenommen. Beispiel:

```
.FI ON;.JU ON
.CH 1
Dies ist die Kapitelüberschrift.
Wenn der Zeilenausgleich eingeschaltet ist, müssen Sie sicherstellen,
dass nach der Kapitelüberschrift ein Zeilenumbruch erfolgt.
.IL 1
Der zwischen der Anweisung .CH und dem nächsten Zeilenumbruch stehende
Text wird in das Inhaltsverzeichnis aufgenommen.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Dies ist die Kapitelüberschrift. Wenn der Zeilenausgleich
   eingeschaltet ist, müssen Sie sicherstellen, dass nach der
   Kapitelüberschrift ein Zeilenumbruch erfolgt.

Der zwischen der Anweisung .CH und dem nächsten Zeilenumbruch stehende
Text wird in das Inhaltsverzeichnis aufgenommen.
```

Das Inhaltsverzeichnis wird automatisch am Ende des formatierten Dokuments ausgegeben:

```
1. Dies ist die Kapitelüberschrift. Wenn der Zeilenausgleich
   eingeschaltet ist, müssen Sie sicherstellen, dass nach der
   Kapitelüberschrift ein Zeilenumbruch erfolgt..... 1
```

Um einen Zeilenumbruch zu definieren (siehe *Eine neue Zeile beginnen*), können Sie entweder Leerzeilen im Rohtext lassen oder das Zeilenendezeichen - normalerweise das Dollarzeichen (\$) - eingeben.

Verschiedene Stufen für Überschriften definieren

Um Überschriften auf verschiedenen Stufen zu definieren, geben Sie die Stufe als Parameter der Anweisung `.CH` an:

Anweisung	Stufe	Nummerierung
<code>.CH 1</code>	Kapitelüberschrift	1
<code>.CH 2</code>	Abschnittsüberschrift	1.1
<code>.CH 3</code>	Überschrift eines Unterabschnitts	1.1.1
<code>.CH 4</code>	Überschrift eines Unterabschnitts eines Unterabschnitts	1.1.1.1



Anmerkung: Der Parameter der Anweisung `.CH` ist nicht die Kapitelnummer, sondern die Stufe der Kapitelüberschrift. Die Kapitelnummer wird automatisch erhöht, sobald die Anweisung `.CH 1` zum nächstenmal ausgeführt wird (d.h. die nächste Überschrift der Stufe 1 definiert wird).

Sie können bis zu zehn Stufen für Überschriften und bis zu 99 Einträge pro Stufe definieren. Jedes Mal, wenn die Anweisung `.CH` ausgeführt wird, wird die Kapitelnummer automatisch vergeben. Die Variable `$CH` enthält die aktuelle Kapitelnummer (siehe [Veränderbare Systemvariablen](#)).

Wenn Sie die Stufe als Parameter der Anweisung `.CH` angeben, werden die Überschriften anfänglich nummeriert. Sie können die Nummerierung jedoch ausschalten (siehe die Anweisung [.OP CHA](#)).

Anfänglich werden die Einträge des Inhaltsverzeichnisses eingerückt. Sie können das Einrücken jedoch ausschalten (siehe die Anweisung [.OP CHI](#)).



Anmerkung: Wenn Sie für Ihre Überschriften keine unterschiedlichen Stufen verwenden möchten, geben Sie die Anweisung `.CH` ohne Parameter an. Die Überschriften werden dann weder nummeriert noch im Inhaltsverzeichnis eingerückt.

Das folgende Beispiel zeigt, wie Überschriften auf verschiedenen Stufen definiert werden:

```
.CH 1
Kapitelüberschrift
Dies ist die Zeile unter der Kapitelüberschrift.
.IL 1
.CH 2
Überschrift des Abschnitts
Dieser Text steht unter der Überschrift der Stufe 2. Im
Inhaltsverzeichnis werden alle Überschriften unter Stufe 1 automatisch
eingerückt.
.IL 1
.CH 3
Überschrift des Unterabschnitts
Dieser Text steht unter der Überschrift der Stufe 3.
.IL 1
.CH 1
Kapitelüberschrift
Dies ist Kapitel 2.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Kapitelüberschrift
Dies ist die Zeile unter der Kapitelüberschrift.

1.1 Überschrift des Abschnitts
Dieser Text steht unter der Überschrift der Stufe 2. Im
Inhaltsverzeichnis werden alle Überschriften unter Stufe 1 automatisch
eingerückt.

1.1.1 Überschrift des Unterabschnitts
Dieser Text steht unter der Überschrift der Stufe 3.

2. Kapitelüberschrift
Dies ist Kapitel 2.
```

Das Inhaltsverzeichnis wird folgendermaßen ausgegeben:

1. Kapitelüberschrift.....	1
1.1 Überschrift des Abschnitts.....	1
1.1.1 Überschrift des Unterabschnitts.....	1
2. Kapitelüberschrift.....	1

Stufen relativ zur aktuellen Stufe definieren

Sie können beim Parameter der Anweisung `.CH` ein Plus- oder Minuszeichen angeben, um eine Stufennummer über oder unter der aktuellen Stufe zu definieren. Wenn Sie `.CH +0` angeben, hat die nächste Überschrift die gleiche Stufe wie die aktuelle Überschrift.

Das folgende Beispiel definiert Überschriften auf verschiedenen Stufen:

```
.CH 1
Erstes Kapitel
Um die nächste Kapitelüberschrift auf der gleichen Stufe zu definieren,
geben Sie .CH +0 an.
.IL 1
.CH +0
Zweites Kapitel
Um eine Abschnittsüberschrift innerhalb des zweiten Kapitels zu
definieren, geben Sie die höhere oder niedrigere Stufe an.
.IL 1
.CH +1
Dies ist die Überschrift des Abschnitts
.CH +1
Ein Unterabschnitt
.CH -1
Eine weitere Überschrift eines Abschnitts
.CH +1
Erster Unterabschnitt
.CH +0
Zweiter Unterabschnitt
.CH +0
Dritter Unterabschnitt
.IL 1
.CH 1
Drittes Kapitel
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Erstes Kapitel
Um die nächste Kapitelüberschrift auf der gleichen Stufe zu definieren,
geben Sie .CH +0 an.

2. Zweites Kapitel
Um eine Abschnittsüberschrift innerhalb des zweiten Kapitels zu
definieren, geben Sie die höhere oder niedrigere Stufe an.

2.1 Dies ist die Überschrift des Abschnitts
2.1.1 Ein Unterabschnitt
2.2 Eine weitere Überschrift eines Abschnitts
2.2.1 Erster Unterabschnitt
2.2.2 Zweiter Unterabschnitt
```

2.2.3 Dritter Unterabschnitt

3. Drittes Kapitel

Das Inhaltsverzeichnis wird folgendermaßen ausgegeben:

1. Erstes Kapitel.....	1
2. Zweites Kapitel.....	1
2.1 Dies ist die Überschrift des Abschnitts.....	1
2.1.1 Ein Unterabschnitt.....	1
2.2 Eine weitere Überschrift eines Abschnitts.....	1
2.2.1 Erster Unterabschnitt.....	1
2.2.2 Zweiter Unterabschnitt.....	1
2.2.3 Dritter Unterabschnitt.....	1
3. Drittes Kapitel.....	1

Eine Überschrift hervorheben

Um eine Überschrift hervorzuheben oder Leerzeilen über oder unter der Überschrift auszugeben, müssen Sie im Rohtext die entsprechenden Anweisungen eingeben. Beispiel:

```
.CH 1;.US;.* Die nächste Zeile wird unterstrichen.
Erstes Kapitel
.IL 1
.CH 1;.BP;.* Die nächste Zeile im Fettdruck ausgeben.
Zweites Kapitel
.* Die nächste Kapitelüberschrift kursiv drucken.
.* Die Umschaltanweisung /I1 wurde in einer extra Zeile eingegeben.
.* Wenn die Umschaltanweisung direkt vor den Wörtern "Drittes Kapitel"
.* eingegeben wird, wird die Nummerierung nicht kursiv gedruckt.
.OP ESC=/
/I1
.CH 1
Drittes Kapitel/I0
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Erstes Kapitel
2. Zweites Kapitel
3. Drittes Kapitel
```

Die Hervorhebungen gelten nicht für das Inhaltsverzeichnis:

1. Erstes Kapitel.....	1
2. Zweites Kapitel.....	1
3. Drittes Kapitel.....	1

Mit der Anweisung .PT können Sie das Layout des Inhaltsverzeichnisses ändern.

.OP CHA - Kapitelnummerierung

```
.OP CHA=NUM  
.OP CHA=UNN
```

Wenn Sie als Parameter der Anweisung .CH eine Stufennummer angeben, werden die Überschriften anfänglich nummeriert. Dies entspricht der folgenden Anweisung:

```
.OP CHA=NUM
```

Um die Kapitelnummerierung auszuschalten, geben Sie an:

```
.OP CHA=UNN
```

Beispiel:

```
.OP CHA=UNN  
.CH 1  
Kapitel 1  
Dies ist das erste Kapitel.  
.CH 2;.IL 1  
Abschnitt 1.1  
Dies ist der erste Abschnitt.  
.CH 2;.IL 1  
Abschnitt 1.2  
Dies ist der zweite Abschnitt.  
.CH 1;.IL 1  
Kapitel 2  
Dies ist das zweite Kapitel.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Kapitel 1  
Dies ist das erste Kapitel.  
  
Abschnitt 1.1  
Dies ist der erste Abschnitt.  
  
Abschnitt 1.2  
Dies ist der zweite Abschnitt.  
  
Kapitel 2  
Dies ist das zweite Kapitel.
```

Das Inhaltsverzeichnis wird folgendermaßen ausgegeben:

Kapitel 1.....	1
Abschnitt 1.1.....	1
Abschnitt 1.2.....	1
Kapitel 2.....	1

Anfänglich werden die Einträge im Inhaltsverzeichnis der Stufe entsprechend eingerückt. Sie können dies jedoch ausschalten (siehe die Anweisung **.OP CHI**).

.OP CHI - Einrückung im Inhaltsverzeichnis

```
.OP CHI=ON  
.OP CHI=OFF
```

Wenn Sie die Stufennummer als Parameter der Anweisung .CH angeben, werden die Überschriften im Inhaltsverzeichnis anfänglich eingerückt - entsprechend der angegebenen Stufe. Dies entspricht der folgenden Anweisung:

```
.OP CHI=ON
```

Um die Einrückung auszuschalten, geben Sie an:

```
.OP CHI=OFF
```

Beispiel:

```
.OP CHI=OFF  
.CH 1  
Kapitel 1  
Dies ist das erste Kapitel.  
.CH 2;.IL 1  
Erster Abschnitt  
Dies ist der erste Abschnitt.  
.CH 2;.IL 1  
Zweiter Abschnitt  
Dies ist der zweite Abschnitt.  
.CH 1;.IL 1  
Kapitel 2  
Dies ist das zweite Kapitel.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Kapitel 1  
Dies ist das erste Kapitel.  
  
1.1 Erster Abschnitt  
Dies ist der erste Abschnitt.  
  
1.2 Zweiter Abschnitt  
Dies ist der zweite Abschnitt.  
  
2. Kapitel 2  
Dies ist das zweite Kapitel.
```

Das Inhaltsverzeichnis wird folgendermaßen ausgegeben:

1. Kapitel 1.....	1
1.1 Erster Abschnitt.....	1
1.2 Zweiter Abschnitt.....	1
2. Kapitel 2.....	1

.OP CHL - Kapitelstufen im Inhaltsverzeichnis

```
.OP CHL=zah1
```

Wenn Sie die Stufennummer als Parameter der Anweisung `.CH` angeben, können Sie als höchste Stufe 10 angeben. Dies entspricht der anfänglichen Einstellung:

```
.OP CHL=10
```

Sie können angeben, bis zu welcher Stufe die Überschriften in das Inhaltsverzeichnis aufgenommen werden sollen. Alle Überschriften, deren Stufe größer ist als die mit der Anweisung `.OP CHL` angegebene Stufe, erscheinen nicht im Inhaltsverzeichnis. Beispiel:

```
.OP CHL=2
.CH 1
Kapitelüberschrift
Die Kapitelüberschrift auf Stufe 1 wird in das Inhaltsverzeichnis
aufgenommen.
.CH 2;.IL 1
Abschnittsüberschrift
Die Überschrift auf Stufe 2 wird in das Inhaltsverzeichnis aufgenommen.
.CH 3;.IL 1
Überschrift des Unterabschnitts
Die Überschrift auf Stufe 3 wird nicht in das Inhaltsverzeichnis
aufgenommen.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Kapitelüberschrift
Die Kapitelüberschrift auf Stufe 1 wird in das Inhaltsverzeichnis
aufgenommen.

1.1 Abschnittsüberschrift
Die Überschrift auf Stufe 2 wird in das Inhaltsverzeichnis aufgenommen.

1.1.1 Überschrift des Unterabschnitts
Die Überschrift auf Stufe 3 wird nicht in das Inhaltsverzeichnis
aufgenommen.
```

Das Inhaltsverzeichnis wird folgendermaßen ausgegeben:

1. Kapitelüberschrift.....	1
1.1 Abschnittsüberschrift.....	1

Wenn Sie überhaupt keine Überschrift in das Inhaltsverzeichnis aufnehmen möchten, geben Sie die folgende Anweisung an:

```
.OP CHI=0
```

Wenn Sie diese Anweisung am Anfang eines Dokuments eingeben, wird kein Inhaltsverzeichnis angelegt. Wenn Sie die Anweisung angeben, nachdem Sie bereits Überschriften in das Inhaltsverzeichnis aufgenommen haben, werden die Überschriften nach dieser Anweisung nicht in das Inhaltsverzeichnis gestellt.

.SC - Kapitelnummer setzen

```
.SC kapitelnummer
```

Diese Anweisung ist nur gültig, wenn die Kapitelnummerierung eingeschaltet ist, d.h. wenn die Anweisung `.OP CHA=NUM` wirksam ist.

Sie können die Nummer des aktuellen Kapitels oder Abschnitts ändern. Die nächste `.CH n` Anweisung bestimmt dann die Stufe des Kapitels oder Abschnitts. Beispiel:

```
.CH 1
Kapitel 1
.SC 4;.** Die aktuelle Kapitelnummer auf 4 setzen.
.CH 1;.** Ein neues Kapitel beginnen. Die Kapitelnummer um 1 erhöhen.
Kapitel 5
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Kapitel 1
5. Kapitel 5
```

Das Inhaltsverzeichnis wird folgendermaßen ausgegeben:

```
1. Kapitel 1..... 1
5. Kapitel 5..... 1
```

Solange Sie keine neue Kapitelüberschrift auf Stufe 1 definieren (mit der Anweisung `.CH 1`), erhalten alle Abschnitte und Unterabschnitte die Kapitelnummer, die dem aktuellen Kapitel zugewiesen wurde. Wenn die Anweisung `.CH 1` nach der Anweisung `.SC` steht, erhöht sich die Kapitelnummer um eins. Beispiel:

```
.CH 1
Kapitelüberschrift
.SC 4;.** Die aktuelle Kapitelnummer auf 4 setzen.
.CH 2;.** Eine Abschnittsüberschrift auf Stufe 2 ausgeben.
Abschnittsüberschrift
.SC 7;.** Die aktuelle Kapitelnummer auf 7 setzen.
.CH 2;.** Eine weitere Abschnittsüberschrift auf Stufe 2 ausgeben.
Abschnittsüberschrift
.CH 3;.** Die Überschrift eines Unterabschnitts auf Stufe 3 ausgeben.
Überschrift des ersten Unterabschnitts
.SC 7.5;.** Die aktuelle Abschnittsnummer auf 7.5 setzen.
.CH 3;.** Eine weitere Überschrift eines Unterabschnitts auf
.** Stufe 3 ausgeben.
Überschrift des zweiten Unterabschnitts
```

```
.SC 15;.** Die aktuelle Kapitelnummer auf 15 setzen.  
.CH 1;.** Ein neues Kapitel mit Nummer 16 beginnen.  
Kapitelüberschrift
```


Die formatierte Ausgabe sieht folgendermaßen aus:

```
1. Kapitelüberschrift
4.1 Abschnittsüberschrift
7.1 Abschnittsüberschrift
7.1.1 Überschrift des ersten Unterabschnitts
7.5.1 Überschrift des zweiten Unterabschnitts
16. Kapitelüberschrift
```

Im Inhaltsverzeichnis werden die Überschriften ihrer Stufe entsprechend eingerückt:

1. Kapitelüberschrift.....	1
4.1 Abschnittsüberschrift.....	1
7.1 Abschnittsüberschrift.....	1
7.1.1 Überschrift des ersten Unterabschnitts.....	1
7.5.1 Überschrift des zweiten Unterabschnitts.....	1
16. Kapitelüberschrift.....	1

.PT - Eintrag in das Inhaltsverzeichnis

```
.PT anweisung  
.PT text
```

Mit dieser Anweisung können Sie das Layout des Inhaltsverzeichnisses steuern. Als Parameter können Sie entweder Text oder eine Anweisung (oder ein Makro mit Parametern) angeben. Es ist nicht möglich, in einer .PT-Anweisung beides zu mischen.

Um z.B. eine zentrierte, unterstrichene Überschrift über dem Inhaltsverzeichnis zu drucken, geben Sie Folgendes an:

```
.PT .CE 1;.US  
.PT Inhaltsverzeichnis
```

Sie können die Anweisung **.HL** (Definition einer Kopfzeile, die auf jeder Seite erscheint) als Parameter der Anweisung .PT angeben. Wie die Anweisung .PT kann auch die Anweisung .HL entweder Text oder eine Anweisung als Parameter haben. Beispiel:

```
.PT .HL .CE 1  
.PT .HL Inhaltsverzeichnis
```

Die Anweisung .PT muss am Anfang des Dokuments stehen. Wenn Sie sie z.B. am Ende des Dokuments eingeben, wird die Ausgabe möglicherweise nicht wie gewünscht formatiert.

Das Inhaltsverzeichnis wird immer am Ende des Dokuments ausgegeben. Wenn Sie die gedruckten Seiten selbst per Hand anders anordnen möchten, so dass das Dokument mit dem Inhaltsverzeichnis beginnt, ist es sinnvoll, für das Inhaltsverzeichnis eine andere Art der Seitennummerierung zu verwenden (z.B. römische Seitenzahlen).

Wenn Sie eine Titelzeile und/oder Fußzeile für das Dokument angeben, gelten diese auch für das Inhaltsverzeichnis, sofern Sie keine andere Titel-/Fußzeile für das Inhaltsverzeichnis definieren. Wenn die Seitennummerierung eingeschaltet ist, werden die Seiten nach der letzten Seite des eigentlichen Dokuments weiter fortlaufend durchnummeriert.



Tipp: Wenn Sie in Con-nect arbeiten, können Sie die formatierte Version des Dokuments ablegen und anschließend das Inhaltsverzeichnis im Con-nect-Editor ändern.

Das folgende Beispiel gibt die Seitenzahl in der Fußzeile des Inhaltsverzeichnisses aus und definiert römische Seitenzahlen in Großbuchstaben, wobei die Seitennummerierung mit eins beginnt.

```
.FI ON;.JU ON
.PT .NP 1
.PT .PM R
.PT .BT //Inh #
.PT .US;.** Die nächste Zeile unterstreichen.
.PT Inhaltsverzeichnis
.PT .IL 1
.CH 1;.** Die Kapitelnummerierung ist anfänglich eingeschaltet.
Das Inhaltsverzeichnis
.IL 1
.CH 2
Die Anweisung .PT
.IL 1
Sie benutzen die Anweisung .PT, um das Layout des Inhaltsverzeichnisses
zu beeinflussen. Mit der Anweisung .PT können Sie Text oder Anweisungen
angeben. Sie können jedoch nicht beides in einer .PT-Anweisung mischen.
.IL 1
.CH 2
Die Anweisung .OP PTC
.IL 1
Die Anweisung .PT ist anfänglich eingeschaltet. Sie können sie
ausschalten, damit der Text oder die Anweisung nach der Anweisung .PT
nicht auf das Inhaltsverzeichnis angewandt wird.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

1. Das Inhaltsverzeichnis

1.1 Die Anweisung .PT

Sie benutzen die Anweisung .PT, um das Layout des Inhaltsverzeichnisses zu beeinflussen. Mit der Anweisung .PT können Sie Text oder Anweisungen angeben. Sie können jedoch nicht beides in einer .PT-Anweisung mischen.

1.2 Die Anweisung .OP PTC

Die Anweisung .PT ist anfänglich eingeschaltet. Sie können sie ausschalten, damit der Text oder die Anweisung nach der Anweisung .PT nicht auf das Inhaltsverzeichnis angewandt wird.

Das Inhaltsverzeichnis wird folgendermaßen ausgegeben:

Inhaltsverzeichnis

1. Das Inhaltsverzeichnis.....	1
1.1 Die Anweisung .PT.....	1
1.2 Die Anweisung .OP PTC.....	1

Die Seitenzahl steht in der Fußzeile des Inhaltsverzeichnisses:

Inh I

.OP PTC - Anweisung .PT ein-/ausschalten

```
.OP PTC=ON  
.OP PTC=OFF
```

Anfänglich ist die Anweisung .PT eingeschaltet, d.h. der Parameter der Anweisung .PT (Text oder Anweisung) gilt auch für das Inhaltsverzeichnis. Dies entspricht der folgenden Anweisung:

```
.OP PTC=ON
```

Um diese Funktion auszuschalten, so dass der Parameter der Anweisung .PT (Text oder Anweisung) nicht auf das Inhaltsverzeichnis angewandt wird, geben Sie ein:

```
.OP PTC=OFF
```

Indexeinträge und Index

.IX - Indexeintrag

```
.IX text
```

Diese Anweisung definiert einen Indexeintrag. Beispiel:

```
.IX Indexeintrag;; definieren
```

Ein Indexeintrag kann bis zu 60 Zeichen lang sein und jedes beliebige druckbare Zeichen enthalten. Wenn Sie das Anweisungstrennzeichen (anfänglich das Semikolon) im Indexeintrag verwenden möchten, müssen Sie es doppelt eingeben (;:).

Die Variable \$IX enthält den mit der letzten .IX-Anweisung angegebenen Text (siehe [Veränderbare Systemvariablen](#)).

Die Anweisung .IX gibt den Indexeintrag nicht an der aktuellen Position im Text aus. Sie speichert den Eintrag zusammen mit der Seitenzahl, auf der der Eintrag definiert wurde.

Die Indexeinträge werden automatisch alphabetisch sortiert und am Ende des Dokuments ausgegeben, auf einer neuen Seite beginnend. Wenn das Dokument auch ein Inhaltsverzeichnis enthält, wird der Index zwischen dem Dokumenttext und dem Inhaltsverzeichnis ausgegeben.

Mit Kleinbuchstaben beginnende Einträge stehen im Index vor großgeschriebenen Einträgen, d.h. die Sortierreihenfolge ist "aAbBcC ... zZ".



Anmerkung: Bei der Installation von Con-form kann die Sortierreihenfolge bestimmt werden.

Wenn Sie die Anweisung .CH angegeben haben, erhält der Index automatisch die Überschrift "Index". Wenn die Kapitelnummerierung eingeschaltet ist (.OP CHA=NUM), wird diese Überschrift mit der nächsten Kapitelnummer versehen. Die Überschrift "Index" erscheint nicht, wenn die Anweisung .CH nicht im Rohtext vorkommt.

"Index *n*" (wobei *n* die Seitenzahl ist) erscheint automatisch unten auf allen ungeraden Seiten. Sie sollten deshalb die Anweisung .BT nicht verwenden, um Seitenzahlen in der Fußzeile zu definieren. Ab der letzten Seite des eigentlichen Dokuments wird weiter fortlaufend nummeriert, - selbst wenn Sie keine Seitenzahlen für Ihr Dokument definiert haben. Auf geraden Seiten wird keine Seitenzahl ausgegeben.

Beispiel

Dieses Beispiel zeigt, wie Indexeinträge definiert werden. Der Zeilenausgleich ist eingeschaltet. Aus diesem Grund bewirken die Indexeinträge keinen Zeilenumbruch im formatierten Dokument. Der Rohtext enthält die Anweisung `.CH`, d.h. der Index erhält eine Kapitelnummer, und es wird ein Inhaltsverzeichnis erstellt. In diesem Beispiel erscheint der Index auf einer geraden Seite, d.h. auf dieser Seite wird keine Seitenzahl ausgegeben.



Anmerkung: Wenn Sie in Con-nect arbeiten und die formatierte Version dieses Beispiels ablegen, ist das Dokumentformat `Cnf` (anstatt `Txt`). Der Grund hierfür ist, dass der Index einen Eintrag enthält, der mit einem Punkt beginnt (`.IX`).

Rohtext

```
.FI ON;.JU ON
.CH 1
Indexeinträge definieren
.IL 1
Ein Indexeintrag wird mit der Anweisung .IX definiert.
.IX .IX-Anweisung
Der Text eines Indexeintrags kann bis zu 60 Zeichen lang sein
.IX Restriktionen für Indexeinträge
und beliebige druckbare Zeichen enthalten.
.IX in Indexeinträgen erlaubte Zeichen
Alle Indexeinträge werden alphabetisch sortiert und am Ende des
Dokuments ausgegeben.
.IX Sortierreihenfolge
```

Formatierte Ausgabe: Dokumenttext

1. Indexeinträge definieren

Ein Indexeintrag wird mit der Anweisung `.IX` definiert. Der Text eines Indexeintrags kann bis zu 60 Zeichen lang sein und beliebige druckbare Zeichen enthalten. Alle Indexeinträge werden alphabetisch sortiert und am Ende des Dokuments ausgegeben.

Formatierte Ausgabe: Index

2. Index

```
in Indexeinträgen erlaubte Zeichen ... 1
Restriktionen für Indexeinträge ... 1
```

Sortierreihenfolge ... 1

.IX-Anweisung ... 1

Formatierte Ausgabe: Inhaltsverzeichnis

1. Indexeinträge definieren.....	1
2. Index.....	2

9 Variablen

▪ Eigene Variablen definieren	206
▪ Arithmetische Berechnungen	213
▪ Textfunktionen	219
▪ Dezimalzahlen	222
▪ Das Variablenzeichen	227
▪ Systemvariablen	229

Dieses Kapitel behandelt die folgenden Themen:

Eigene Variablen definieren

Wenn ein langes Wort oder Satzteil im Rohtext häufig vorkommt, können Sie es einer Variablen zuweisen. Anschließend brauchen Sie nur noch die Variable anzugeben.

.SV - Variable definieren

```
.SV variablenname=wert
```

Mit dieser Anweisung definieren Sie eine Variable und weisen ihr einen Wert zu. Beispiel:

```
.SV Name=Schmitt
```

Jede Variable muss einen eindeutigen Namen haben. Ein Variablenname (der Parameter vor dem Gleichheitszeichen) kann nur Buchstaben und Zahlen enthalten. Er muss mindestens ein Zeichen und kann höchstens 100 Zeichen lang sein. Der Variablenname darf nicht mit dem Variablenzeichen (siehe unten) beginnen und darf keine Leerzeichen enthalten. Zwischen Groß- und Kleinschreibung wird nicht unterschieden.

Der Parameter nach dem Gleichheitszeichen kann entweder ein Text (kann aus mehreren Wörtern bestehen und kann Leerzeichen enthalten) oder eine arithmetische Berechnung sein (siehe [Arithmetische Berechnungen mit .SV](#)).

	Maximaler Wert für Text	Maximaler Wert für eine arithmetische Berechnung
Parameter nach dem Gleichheitszeichen	249 Zeichen	100 Zeichen
Zwischenergebnis	CSIZE minus 10 (in KB)	31 Stellen
Endergebnis	CSIZE minus 10 (in KB)	29 Stellen

 **Anmerkungen:**

1. Ab Con-form Version 3.4.1 darf der maximale Wert für Text 100 Zeichen überschreiten. In vorhergehenden Versionen war dies auf 100 Zeichen beschränkt.
2. CSIZE ist die Größe des Con-nect-Buffers. Der Administrator definiert dies im Natural-Parametermodul NATPARM. Der maximale Wert für CSIZE kann 512 KB betragen. Somit können 502 KB als Maximum für das Zwischenergebnis und das Endergebnis benutzt werden.

Zwischen den Variablen, die Text enthalten, und Variablen, die numerische Werte enthalten, wird kein Unterschied gemacht. Eine Variable kann zu einem Zeitpunkt Text enthalten und später einen

numerischen Wert. Der Typ des aktuellen Wertes einer Variablen wird durch die letzte .SV-Anweisung bestimmt, die dieser Variablen einen Wert zuwies.

Eine Variable im Dokumenttext benutzen

Nachdem Sie eine Variable mit der Anweisung `.SV` definiert haben, können Sie sie im Rohtext verwenden. Vor der Variablen müssen Sie das Variablenzeichen angeben. Anfänglich ist das Variablenzeichen das kaufmännische Und (`&`). Beispiel:

```
&Name
```

Beim Formatieren des Dokuments wird die Variable durch den definierten Wert ersetzt.

Die Ersetzung (siehe die Anweisung `.SU`) wird von der Anweisung `.SV` automatisch eingeschaltet. Das bedeutet, dass der Text, der dem Variablenzeichen folgt, als Variable interpretiert und durch den entsprechenden Wert ersetzt wird.

Textzuweisungen

Sie können beliebig viele Variablen und Textzuweisungen definieren.

Um eine Zeichenkette oder eine andere Variable direkt nach der Variablen auszugeben (ohne Leerzeichen dazwischen), müssen Sie nach der ersten Variablen einen Punkt (`.`) angeben. Der Punkt ist nicht erforderlich nach einer Zeichenkette, auf die eine Variable folgt. Beispiel:

```
.SV var1=Blumen  
.SV var2=kohl  
.SV dm=200  
&var1.&var2  
&var1.strauß  
Rosen&var2  
&dm DM
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Blumenkohl  
Blumenstrauß  
Rosenkohl  
200 DM
```

Zusammengesetzte Variablennamen

Eine Variable kann gebildet werden, indem ein zusammengesetzter Name aufgelöst wird.

Ein zusammengesetzter Name enthält zwei oder mehrere einfache Variablen und wird aufgelöst, indem der Wert jeder Variablen eingesetzt wird. Dabei wird von rechts nach links vorgegangen.

Beispiel:

```
.SV V1wecken=Guten Morgen  
.SV V2Nacht=wecken  
.SV V3=Nacht  
&V1&V2&V3
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Guten Morgen
```

Die Variable im obigen Beispiel wird von rechts nach links in aufeinander folgenden Schritten aufgelöst:

```
&V1&V2&V3  
&V1&V2Nacht  
V1wecken  
Guten Morgen
```

Satzzeichen nach einer Variablen

Um nach einer Variablen ein Satzzeichen auszugeben, müssen Sie einen Punkt (.) gefolgt vom gewünschten Satzzeichen eingeben. Um z.B. einen Satz mit einem Punkt abzuschließen, müssen Sie den Punkt doppelt angeben:

```
.SV text=In diesem Fall müssen Sie den Punkt wiederholen  
&text..  
.SV bsp=Diesem Text folgt ein Semikolon  
&bsp.;
```

Die formatierte Ausgabe sieht folgendermaßen aus:

In diesem Fall müssen Sie den Punkt wiederholen.
Diesem Text folgt ein Semikolon;

Arithmetische Berechnungen mit .SV

Der Parameter der Anweisung .SV kann ein numerischer Wert sein. Sie können eine ganze Zahl, eine gebrochene Zahl mit Dezimalzeichen oder eine Zahl mit einem Tausendertrennzeichen angeben (siehe die Anweisung [.OP TRI](#)).

Sie können die folgenden arithmetischen Operatoren benutzen:

+	Addition
-	Subtraktion
*	Multiplikation
/	Division

Die Anweisung .SV führt die Berechnung von links nach rechts aus. Alle Operatoren haben die gleiche Priorität. Klammern werden nicht ausgewertet.



Tipp: Es wird empfohlen, dass Sie bei arithmetischen Berechnungen die Anweisung .CV benutzen. Mit dieser Anweisung werden Klammern ausgewertet. Siehe [.CV - Variable berechnen](#).

Sie können gebrochene Zahlen angeben. Wenn Sie jedoch die Anzahl der Stellen nach dem Dezimalzeichen nicht definieren (siehe die Anweisung [.OP DAS](#)), wird das Ergebnis auf die nächste ganze Zahl gerundet. Beispiel:

```
.SV zahl1=2.5*3+2
Die Variable zahl1 hat den Wert &zahl1
.OP DAS=2
.SV zahl2=2.5*3+2
Die Variable zahl2 hat den Wert &zahl2
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Die Variable zahl1 hat den Wert 9
Die Variable zahl2 hat den Wert 9.50
```



Vorsicht: Sie müssen die Anweisung .OP DAS vor der Anweisung .SV angeben. Andernfalls wird das Ergebnis nicht wie gewünscht gerundet.

Sie können den Wert eines Parameters ändern. Beispiel:

```
.SV zähler=11  
.SV zähler=&zähler+1  
&zähler
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
12
```

Anfänglich wird das Divisionsergebnis auf die nächste ganze Zahl gerundet. Wenn Sie jedoch angeben, wie viele Stellen nach dem Dezimalzeichen ausgegeben werden sollen (siehe die Anweisung **.OP DAS**), wird das Ergebnis entsprechend gerundet. Beispiel:

```
.OP DAS=2  
.SV zahl=2/3  
&zahl
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
0.67
```


Arithmetische Berechnungen

Es wird empfohlen, dass Sie bei arithmetischen Berechnungen die Anweisung `.CV` benutzen. Im Gegensatz zur Anweisung `.SV` werden hiermit Klammern ausgewertet.

`.CV` - Variable berechnen

```
.CV variablenname=arithmetischer-ausdruck
```

Sie können eine Variable definieren und deren Wert anschließend mit einem arithmetischen Ausdruck berechnen. Ein arithmetischer Ausdruck besteht aus einer oder mehreren Konstanten, Variablen oder Systemvariablen.

Eine Konstante ist eine ganze Zahl, eine gebrochene Zahl mit Dezimalzeichen, oder eine Zahl mit einem Tausendertrennzeichen (siehe die Anweisung `.OP TRI`). Arithmetische Operatoren (siehe unten) dürfen nicht als Tausendertrennzeichen verwendet werden. Eine Konstante darf nicht mehr als 29 Stellen haben. Nach dem Dezimalzeichen dürfen nicht mehr als 7 Stellen stehen. Beispiel:

```
.OP TRI=', '
.CV const1=15
.CV const2=0.123
.CV const3=1234567890123456789012.1234567
.CV const4=1,999
```

Eine Variable in einem arithmetischen Ausdruck muss einen numerischen Wert haben. Diese Variable muss vorher mit der Anweisung `.SV` oder `.CV` definiert werden.

Eine Systemvariable kann in einem arithmetischen Ausdruck verwendet werden, wenn sie einen numerischen Wert hat.

Sie können die folgenden arithmetischen Operatoren benutzen:

()	Klammern
*	Multiplikation
/	Division
+	Addition
-	Subtraktion

Klammern werden ausgewertet. Eine arithmetische Operation wird in folgender Reihenfolge ausgeführt:

1. Klammern

2. Multiplikation und Division (von links nach rechts)
3. Addition und Subtraktion (von links nach rechts)

Wenn der Divisor 0 ist, ist das Divisionsergebnis 0.

Wenn das Ergebnis einer Berechnung mit .CV zu einem Datenüberlauf führt, wird die Variable mit 5 Sternen (*****) gefüllt. Beispiel:

```
.CV var1=-25000
&var1
.CV var2=15-5+4/2-10
&var2
.CV var3=(1+2) * 3/ (1+8)
&var3
.CV var4=&var1+100000
&var4
.CV var5= 10000000000000000 * 10000000000000000
.IF &var5 = *****
.TH
Ihr Ergebnis ist zu groß
.EL
&var5
.EI
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
-25000
2
1
75000
Ihr Ergebnis ist zu groß
```

Genauigkeit der Resulte von Arithmetischen Operationen

Die folgende Tabelle gibt eine Übersicht über die Genauigkeitsregeln, die in einer arithmetischen Berechnung mit .CV verwendet werden.

Operation	Stellen vor Dezimalzeichen	Stellen nach Dezimalzeichen (Zwischenergebnis)	Stellen nach Dezimalzeichen (Endergebnis)
Addition/Subtraktion	$F_i + 1$ oder $S_i + 1$ (je nachdem was größer ist)	F_d oder S_d (je nachdem was größer ist, Maximum 7)	DAS und RND werden beim letzten
Multiplikation	$F_i + S_i + 2$	$F_d + S_d$ (Maximum 7)	Zwischenergebnis angewendet.
Division	$F_i + S_d$	Bei .OP REM=OFF: F_d oder Wert der Option DAS (je nachdem was größer ist, Maximum 7). Bei .OP RND=ON wird diese Anzahl der Stellen zusätzlich intern erhöht (+1).	
		Bei .OP REM=ON: Wert der Option DAS. Dieser Wert wird auch für den Rest benutzt.	

In dieser Tabelle werden folgende Abkürzungen benutzt:

F	Erster Operand.
S	Zweiter Operand.
i	Stellen vor Dezimalzeichen.
d	Stellen nach Dezimalzeichen.

Die Stellen nach dem Dezimalzeichen werden bei Zwischen- und Endergebnis unterschiedlich behandelt. Zwischenergebnisse sind arithmetische Operationen *vor* Runden und Anwenden der Option DAS. Das Endergebnis ist das letzte Zwischenergebnis *nach* Runden und Anwenden der Option DAS.

Beim Endergebnis (formatierte Ausgabe) werden die Stellen nach dem Dezimalzeichen durch die Option DAS bestimmt.

Bei .OP RND=ON wird gerundet. Bei .OP RND=OFF werden die letzten Stellen abgeschnitten.

Bei .OP REM=ON wird das Divisionsergebnis nicht gerundet (unabhängig von der Einstellung für .OP RND).

Beispiel:

```
.OP DAS=0
.OP RND=ON
.CV VAR=(3+5.25) / 1.5 - 4
```

Das Endergebnis dieser Berechnung ist 2. Die jeweiligen Zwischenergebnisse können Sie der folgenden Tabelle entnehmen:

	Operation	Ergebnis
Zwischenergebnis 1	3+5.25	8.25
Zwischenergebnis 2	8.25/1.5	5.500 (wegen .OP RND=ON wird die Anzahl der Stellen nach dem Dezimalzeichen intern erhöht (+1)).
Zwischenergebnis 3	5.500-4	1.500
Endergebnis	DAS und RND anwenden	2

.OP RND - Ergebnis von .CV runden

```
.OP RND=ON  
.OP RND=OFF
```

Diese Option kann nur zusammen mit .CV verwendet werden.

Anfänglich wird das Ergebnis von .CV auf die nächste ganze Zahl aufgerundet. Dies entspricht der Anweisung:

```
.OP RND=ON
```

Beispiel:

```
.OP DAS=3  
.OP RND=ON  
.CV var1=0 + 0.2555  
&var1  
.OP RND=OFF  
.CV var2=0 + 0.2555  
&var2
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
0.256  
0.255
```

.OP REM - Rest bei Division mit .CV

```
.OP REM=ON  
.OP REM=OFF
```

Diese Option kann nur zusammen mit `.CV` verwendet werden.

Anfänglich ist diese Option ausgeschaltet. Dies entspricht der Anweisung:

```
.OP REM=OFF
```

Wenn der Divisionsrest an die veränderbare Systemvariable `$RR` übergeben werden soll, müssen Sie Folgendes angeben:

```
.OP REM=ON
```

Der anfängliche Wert von `$RR` ist 0 (Null).

Das Divisionsergebnis wird nicht gerundet (unabhängig von der Einstellung für `.OP RND`). Das Restformat ist abhängig von den Optionen, die mit den Anweisungen `.OP DAS` und `.OP TRI` angegeben wurden.

Beispiel:

```
.OP DAS=2  
.OP REM=ON  
&$RR  
.CV zah1=2/3  
&zah1  
&$RR
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
0  
0.66  
0.02
```

Textfunktionen

In Verbindung mit der Anweisung `.SV` oder `.CV` können Sie Textfunktionen verwenden. Nach dem Funktionscode müssen Sie immer einen Apostroph (') angeben.

Textfunktionen können auch mit den Anweisungen `.IF` und `.WH` benutzt werden.

Es gibt folgende Textfunktionen:

Code	Erklärung	Beispiel
E	Prüfen, ob die angegebene Variable existiert. Der Parameter ist der Name einer Variablen. Wenn die Variable existiert, wird der Wert 1 ausgegeben; wenn sie nicht existiert, der Wert 0.	<code>.SV var=Schmitt</code> <code>.SV test=var</code> Wenn die Variable "var" existiert, wird 1, andernfalls 0 ausgegeben: &E'&test
H	Eine Zeichenkette in einen Hexadezimalwert konvertieren. Der Parameter kann eine beliebige Zeichenkette sein. Jedes darin enthaltene Zeichen wird in den entsprechenden EBCDIC Hexadezimalwert konvertiert.	<code>.SV var=H'123</code> Der hexadezimale Wert von 123 ist &var. <code>.SV name=Schmitt</code> Der hexadezimale Wert von Schmitt ist &H'&name.
I	Die Richtung, in der die Zeichenkette verarbeitet wird, ändern. Der Parameter ist eine beliebige Zeichenkette. Die Richtung, in der der Inhalt der Variablen interpretiert wird, wird geändert.	<code>.SV abc=xyz</code> <code>.SV def=I'&abc</code> Der invertierte Wert ist &def (zyx).
L	Eine Zeichenkette in Kleinbuchstaben umwandeln. Der Parameter ist eine beliebige Zeichenkette. Jeder Großbuchstabe in der Zeichenkette wird durch den entsprechenden Kleinbuchstaben ersetzt. Andere Zeichen bleiben unverändert.	<code>.SV var=HEUTE</code> Nicht vergessen: &L'&var ist ein Feiertag.
N	Die Länge einer Zeichenkette bestimmen. Der Parameter ist eine beliebige Zeichenkette. Die Anzahl der Zeichen wird ausgegeben.	<code>.SV Datum=3.11.00</code> Die Länge dieser Zeichenkette ist &N'&Datum
R	Einen Parameter in eine römische Zahl konvertieren. Der Parameter ist eine Dezimalzahl ohne Vorzeichen. Die entsprechende römische Zahl in Großbuchstaben wird ausgegeben.	<code>.SV Jahr=R'2000</code> &Jahr wird ein wundervolles Jahr werden. <code>.SV Band=8</code> Gib mir Band &R'&Band der Enzyklopädie.

Code	Erklärung	Beispiel
U	Eine Zeichenkette in Großbuchstaben umwandeln. Der Parameter ist eine beliebige Zeichenkette. Jeder Kleinbuchstabe in der Zeichenkette wird durch den entsprechenden Großbuchstaben ersetzt. Andere Zeichen bleiben unverändert.	<pre>.SV var=Schmitt Bitte rufen Sie dringend Herrn &U'&var an.</pre>
X	Einen Hexadezimalwert in ein Zeichen konvertieren. Der Parameter muss ein einstelliger Hexadezimalwert sein. Er wird in das entsprechende Zeichen konvertiert.	<pre>.SV hex=C5 Der hexadezimale Wert C5 entspricht dem Wert &X'&hex .SV hexa=X'C5 Der hexadezimale Wert C5 entspricht dem Wert &hexa</pre>

Eine Textfunktion als Parameter angeben

Sie können eine Textfunktion als Parameter der Anweisung `.SV` oder `.CV` angeben. Dazu müssen Sie den entsprechenden Funktionscode gefolgt vom Parameter angeben. Sie können dann die Variable in den laufenden Text einfügen; direkt vor der Variablen muss das Variablenzeichen stehen. Anfänglich ist das Variablenzeichen das kaufmännische Und (`&`). Beispiel:

```
.SV var=H'123  
Der hexadezimale Wert von 123 ist &var
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Der hexadezimale Wert von 123 ist F1F2F3
```

Eine Textfunktion im laufenden Text angeben

Wenn Sie eine Textfunktion im laufenden Text angeben, müssen Sie vor dem Funktionscode das Variablenzeichen angeben. Außerdem müssen Sie nach der Textfunktion eine definierte Variable angeben. Auch vor der Variablen muss das Variablenzeichen stehen. Anfänglich ist das Variablenzeichen das kaufmännische Und (`&`). Beispiel:

```
.SV var=123  
Der hexadezimale Wert von 123 ist &H'&var
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Der hexadezimale Wert von 123 ist F1F2F3
```

Dezimalzahlen

Mit den folgenden Optionen können Sie angeben, wie Dezimalzahlen ausgegeben werden sollen, wenn mit Hilfe von Variablen arithmetische Berechnungen durchgeführt werden.

.OP DAS - Stellen nach Dezimalzeichen

```
.OP DAS=zah1
```

Anfänglich wird das Ergebnis einer arithmetischen Berechnung auf die nächste ganze Zahl auf- oder abgerundet. Um dies zu vermeiden, können Sie angeben, wie viele Stellen nach dem Dezimalzeichen ausgegeben werden sollen. Das Dezimalzeichen wird mit der Anweisung `.OP DEC` definiert (siehe unten).

Um z.B. nach dem Dezimalzeichen 2 Stellen auszugeben, geben Sie an:

```
.OP DAS=2
```

Sie können bis zu 7 Zeichen hinter dem Dezimalzeichen angeben.

Das Dezimalzeichen wird *nicht* automatisch eingefügt, wenn Sie die Anweisung `.SV` oder `.CV` wie im folgenden Beispiel angeben:

```
.OP DAS=2  
.SV zah1=1  
&zah1
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
1
```

Um die obige Variable mit Dezimalzeichen auszugeben, müssen Sie den folgenden zusätzlichen Berechnungsschritt ausführen:

```
.OP DAS=2  
.SV zah1=1+0  
&zah1
```

Die formatierte Ausgabe sieht folgendermaßen aus:

1.00

.OP DEC - Dezimalzeichen definieren

```
.OP DEC=zeichen
```

Anfänglich wird das in Natural definierte Dezimalzeichen benutzt. Standardmäßig ist dies der Punkt (.). Sie können das Dezimalzeichen ändern. Um z.B. das Komma als neues Dezimalzeichen zu definieren, geben Sie an:

```
.OP DEC=' , '
```

Da das Komma bei bestimmten Con-form-Anweisungen als Parametertrennzeichen benutzt wird, ist es wichtig, es wie im obigen Beispiel in Apostrophe einzuschließen. Wenn Sie z.B. den Schrägstrich definieren möchten, brauchen Sie keine Apostrophe anzugeben (.OP DEC=/).

.OP TRI - Tausendertrennzeichen definieren

```
.OP TRI=zeichen
.OP TRI=ON
.OP TRI=OFF
```

Anfänglich ist kein Tausendertrennzeichen definiert.

Sie können das Zeichen angeben, das als Trennzeichen für Tausender benutzt werden soll. Um z.B. das Komma als Tausendertrennzeichen zu definieren, geben Sie an:

```
.OP TRI=','
```

Da das Komma bei bestimmten Con-form-Anweisungen als Parametertrennzeichen benutzt wird, ist es wichtig, es wie im obigen Beispiel in Apostrophe einzuschließen. Wenn Sie z.B. den Schrägstrich definieren möchten, brauchen Sie keine Apostrophe anzugeben (.OP TRI=/).

Das Tausendertrennzeichen wird in der formatierten Ausgabe in Ergebnissen von arithmetischen Berechnungen ausgegeben. Es erscheint *nicht*, wenn das Ergebnis der Anweisung .SV oder .CV wie im folgenden Beispiel ausgegeben wird:

```
.OP TRI=.
.SV zahl=2000
&zahl
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
2000
```

Um diese Zahl mit dem Tausendertrennzeichen auszugeben, müssen Sie einen zusätzlichen Berechnungsschritt ausführen:

```
.OP TRI=.
.SV zahl=2000+0
&zahl
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
2.000
```

Wenn Sie das Tausendertrennzeichen nicht mehr verwenden wollen, können Sie es mit der folgenden Anweisung deaktivieren:

```
.OP TRI=OFF
```

Wenn Sie das vorher definierte, deaktivierte Tausendertrennzeichen wieder verwenden wollen, können Sie es mit der folgenden Anweisung wieder aktivieren:

```
.OP TRI=ON
```

Wenn Sie `.OP TRI=ON` angeben und vorher noch kein Tausendertrennzeichen definiert wurde, wird standardmäßig das Komma verwendet.

Das Variablenzeichen

Das Variablenzeichen muss vor allen Variablen im Rohtext (d.h. Systemvariablen und Variablen, die mit der Anweisung `.SV` oder `.CV` definiert wurden) angegeben werden. Bei Bedarf können Sie auch ein anderes Variablenzeichen definieren oder die Erkennung des Variablenzeichens ausschalten.

.OP VSG - Variablenzeichen definieren

```
.OP VSG=zeichen
```

Diese Anweisung definiert das Zeichen, das benutzt wird, um Text und Variablen zu unterscheiden.

Anfänglich ist das kaufmännische Und (&) das Variablenzeichen. Sie können das Zeichen ändern. Um z.B. das Paragrafenzeichen (§) als Variablenzeichen zu definieren, geben Sie an:

```
.OP VSG=§
```

Es ist nicht möglich, das Dollarzeichen (\$) als Variablenzeichen für Systemvariablen zu definieren. Daher ist es nicht möglich, z.B. `$$PL` anzugeben. Sie können jedoch Folgendes angeben:

```
.SV PL=&§PL
.OP VSG=§
$PL
```

.SU - Ersetzung

```
.SU ON
.SU OFF
```

Wenn die Ersetzung eingeschaltet ist, wird jede Zeichenkette im Rohtext, vor der das Variablenzeichen steht, als Variable interpretiert und entsprechend ersetzt.

Anfänglich ist die Ersetzung ausgeschaltet. Sie wird durch die Anweisung `.SV` oder `.CV` automatisch eingeschaltet. Dies entspricht der Anweisung:

```
.SU ON
```

Die Ersetzung wird auch durch Makroaufrufe *mit* Parametern eingeschaltet. Sie wird nicht eingeschaltet, wenn ein Makro ohne Parameter aufgerufen wird.

Sie können die Ersetzung ausschalten. Dies ist erforderlich, wenn der Rohtext das Variablenzeichen als Text enthalten soll, d.h. wenn das Variablenzeichen nicht als solches interpretiert werden soll.

Um die Ersetzung auszuschalten, geben Sie an:

```
.SU OFF
```

Systemvariablen

Außer den Variablen, die Sie selbst definieren (mit der Anweisung `.SV` oder `.CV`), können Sie im Rohtext auch Systemvariablen verwenden. Systemvariablen enthalten Werte wie z.B. Datum und Uhrzeit oder die aktuellen Ränder des Textes.

Wenn Sie eine Systemvariable angeben, müssen Sie direkt davor das Variablenzeichen eingeben. Anfänglich ist das kaufmännische Und (&) das Variablenzeichen. Um z.B. die Systemvariable `$DT` zu benutzen, geben Sie an:

```
&$DT
```

Sie können eine Variable nicht nur im laufenden Text einfügen, sondern sie auch in den Anweisungen `.IF` und `.WH` benutzen.

Feste Systemvariablen

Es gibt Systemvariablen, die nicht verändert werden können. Con-form ersetzt diese Variablen beim Formatieren des Dokuments automatisch durch ihren aktuellen Wert.

Variable	Erklärung	Wert
\$CN	Jahrhundert (die ersten beiden Zahlen des Jahres)	19 oder 20
\$DA	Tageszahl	1 bis 31
\$DD	Tagesname in Dänisch	Mandag...Søndag
\$DF	Tagesname in Französisch	Lundi...Dimanche
\$DG	Tagesname in Deutsch	Montag...Sonntag
\$DN	Tagesname in Englisch	Monday...Sunday
\$DT	Datum Bei den Monaten 1 bis 9 steht vor der Monatszahl ein Leerzeichen (z.B. 1. 9.94).	tt.mm.jj
\$DY	Julianisches Datum	1 bis 366
\$HO	Stunde	00 bis 23
\$MD	Monatsname in Dänisch	Januar...December
\$MF	Monatsname in Französisch	Janvier...Décembre
\$MG	Monatsname in Deutsch	Januar...Dezember
\$MI	Minute	00 bis 59
\$MN	Monatsname in Englisch	January...December
\$MO	Monatsnummer	1 bis 12
\$SE	Sekunde	00 bis 59
\$YE	Jahr	00 bis 99

Das folgende Beispiel zeigt, wie diese Systemvariablen benutzt werden:

```
Das aktuelle Datum ist: &$DG., &$DA.. &$MG &$CN.&$YE..
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Das aktuelle Datum ist: Mittwoch, 25. Oktober 2000.
```

Veränderbare Systemvariablen

Die folgenden Variablen können für komplexe Formatierungen benutzt werden. Sie sind anfänglich mit vorgegebenen Werten belegt. Mit den Anweisungen, die in der folgenden Tabelle in der rechten Spalte aufgeführt sind, werden die Systemvariablen geändert.

Variable	Erklärung	Geändert durch
\$BM	Leerzeilen nach der Fußzeile	.BM
\$CH	Kapitelnummer	.CH, .SC
\$FM	Abstand zwischen Text und Fußzeile	.FM
\$FN	Fußnotenzähler Die Variable \$FN wird jedes Mal erhöht, wenn sie referenziert wird. Sie dient zur fortlaufenden Nummerierung der Fußnoten.	
\$FS	Zeilenreservierung für Fußzeile	.FS
\$HM	Abstand zwischen Titelzeile und Text	.HM
\$HS	Zeilenreservierung für Titelzeile	.HS
\$IN	Einzug	.LM, .OF, .TI, .CS
\$IX	Letzter Indexeintrag	.IX
\$LC	Restliche Zeilen auf der Seite	.SV, .LS
\$LL	Zeilenlänge	.LL, .CS
\$PL	Seitenlänge	.PL, .LS
\$PN	Aktuelle Seitenzahl. Beim Ersetzen der Variablen \$PN werden arabische Seitenzahlen benutzt. Mit der Anweisung .SV oder .CV sowie in der Titel- und Fußzeile können Sie anstelle von \$PN das Seitenzahlzeichen (anfänglich das Nummernzeichen (#)) benutzen. In diesem Fall wird die Seitenzahl immer Ihren Angaben entsprechend (entweder arabisch oder römisch) ausgegeben.	.NP, .PN
\$RM	Rechter Rand	.RM, .CS
\$RR	Divisionsrest wenn .OP REM=ON	.CV
\$TM	Leerzeilen über der Titelzeile	.TM

Diese Systemvariablen geben Werte zurück, die von Con-form intern gespeichert wurden. Es ist jedoch nicht gewährleistet, dass die Systemvariablen in verschiedenen Umgebungen und mit verschiedenen Versionen von Con-form immer die gleichen Ergebnisse liefern.

Das folgende Beispiel zeigt, wie veränderbare Systemvariablen eingesetzt werden. Es verwendet außerdem die Anweisung **.IF**.

Wenn auf der aktuellen Seite weniger als 5 Zeilen verfügbar sind, wird der Rahmen auf der nächsten Seite ausgegeben. Die aktuelle Seitenzahl wird im Rahmen ausgegeben.

```
.SL 1
.IF &$LC < 5;.NP
.BX 10,55
.LM 10;.RM 54
.IL 1;.CE 1
Dieser Rahmen wird auf Seite &$PN ausgedruckt.
.IL 1;.BX OFF
```

Die formatierte Ausgabe sieht folgendermaßen aus:

Wenn auf der aktuellen Seite weniger als 5 Zeilen verfügbar sind, wird der Rahmen auf der nächsten Seite ausgegeben. Die aktuelle Seitenzahl wird im Rahmen ausgegeben.

```
+-----+
!                                     !
!Dieser Rahmen wird auf Seite 1 ausgedruckt. !
!                                     !
+-----+
```

10 Bedingte und wiederholte Verarbeitung

▪ Bedingte Verarbeitung	235
▪ Wiederholte Verarbeitung	239
▪ Eine Bedingung definieren	242
▪ Eine Konsequenz definieren	245
▪ Logische Operatoren	246

Dieses Kapitel behandelt die folgenden Themen:

Wenn Sie die Anweisung **.EP** in der Anweisung **.IF** verwenden, wird die Formatierung immer beendet.

Bedingte Verarbeitung

.IF - Beginn der bedingten Verarbeitung

```
.IF bedingung
```

Die bedingte Verarbeitung beginnt mit der Anweisung .IF, mit der die Bedingung (siehe *Eine Bedingung definieren*) definiert wird.

Wenn der Vergleich das Ergebnis "wahr" liefert, wird die definierte Konsequenz (siehe *Eine Konsequenz definieren*) verarbeitet. Ist das Ergebnis falsch, wird die definierte Konsequenz übersprungen.

Wenn Sie die Anweisung .IF benutzen, können Sie entweder eine einfache Konsequenz (siehe unten) oder eine Blockkonsequenz (siehe die Anweisung **.TH**) definieren.

Die einfache Konsequenz

Eine einfache Konsequenz besteht aus einer einzigen Textzeile oder einer Anweisung, die auf die Anweisung .IF folgt. Beispiel:

```
.IF &Alter = 50
Herzlichen Glückwunsch
```

Die einfache Konsequenz beendet die bedingte Verarbeitung.



Anmerkung: Nach einer einfachen Konsequenz sollten Sie die Anweisung .EI (End-If) nicht angeben.

Sie können in einer Bedingung auch die logischen Operatoren .AN und .OR (siehe *Logische Operatoren*) benutzen. Im folgenden Beispiel ist die Textzeile "Sehr geehrter Herr" die einfache Konsequenz:

```
.IF &Alter = 50;.AN &Geschlecht = M
Sehr geehrter Herr
```

Wenn die einfache Konsequenz aus einer Anweisung besteht, können Sie sie nach der Anweisung .IF in derselben Zeile angeben. Im folgenden Beispiel ist die Anweisung .NP die einfache Konsequenz:

```
.IF &$LC < 5;.NP
```


.TH - Blockkonsequenz definieren

.TH

Eine Blockkonsequenz besteht aus mehreren Textzeilen und/oder Anweisungen nach der Anweisung .IF. Sie muss mit der Anweisung .TH beginnen und mit der Anweisung .EI enden.
Beispiel:

```
.IF &$DG = Montag
.TH
.NP
&$DA.. &$MG &$YE
.IL 1
Dies ist der Anfang einer neuen Woche.
.EI
```

In diesem Beispiel bilden die Anweisungen und der Text zwischen den Anweisungen .TH und .EI die Blockkonsequenz (eine neue Seite beginnen; das aktuelle Datum ausgeben; eine Leerzeile ausgeben; den Text ausgeben).

Die Anweisungen .TH und .EI bilden ein Paar, d.h. wenn Sie die Anweisung .TH verwenden, müssen Sie auch die dazugehörige .EI-Anweisung eingeben.



Anmerkung: Bedingungen können geschachtelt werden. Die maximale Schachtelungstiefe ist 10.

.EL - Alternative definieren

```
.EL
```

Wenn Sie die Anweisung .TH angeben, können Sie auch eine Alternative dazu definieren. Wenn das Ergebnis der definierten .IF-Bedingung falsch ist, wird die Alternative ausgeführt (anstelle der Blockkonsequenz nach der Anweisung .TH).

Eine Alternative kann aus mehreren Textzeilen und/oder Anweisungen nach der Anweisung .EL bestehen. Beispiel:

```
.IF &Geschlecht = M  
.TH  
Herr  
.EL  
Frau  
.EI
```

Wenn die Variable &Geschlecht in diesem Beispiel nicht den Wert M hat, wird die Alternative zwischen den Anweisungen .EL und .EI ausgeführt (d.h. im formatierten Text wird "Frau" ausgegeben).

Die für die Blockkonsequenz geltenden Regeln gelten auch für die Alternative (siehe [Eine Konsequenz definieren](#)).



Anmerkung: Wenn Sie die Anweisung .EL benutzen, müssen Sie auch die Anweisungen .TH und .EI benutzen.

.EI - Ende der bedingten Verarbeitung

```
.EI
```

Die Anweisung .EI schließt eine Blockkonsequenz (einschließlich der Alternative) ab, die mit der Anweisung .TH eingeleitet wurde.



Anmerkung: Auf eine .TH-Anweisung muss eine .EI-Anweisung folgen.

Die gesamte .IF-Konstruktion, d.h. sämtliche Angaben zwischen den Anweisungen .IF und .EI müssen im selben Dokument angegeben werden.

Wiederholte Verarbeitung

.WH - Schleifenkopf

```
.WH bedingung
```

Mit dieser Anweisung beginnt die wiederholte Verarbeitung. Die Anweisung .WH definiert eine Bedingung (siehe *Eine Bedingung definieren*). Auf die Bedingung folgt eine Konsequenz, die aus mehreren Textzeilen und/oder Anweisungen bestehen kann, die wiederholt werden sollen (siehe *Eine Konsequenz definieren*).

Das Ende der Schleife muss mit der Anweisung .EW definiert werden.

Solange die angegebene Bedingung wahr ist, wird die definierte Konsequenz ausgeführt. Wenn die Bedingung falsch ist, wird die Schleife beendet. Die Verarbeitung fährt nach der Anweisung .EW fort.

Es folgt ein einfaches Beispiel einer Schleife:

```
.SV zähler=0
.WH &zähler <= 3;.** 3fache Wiederholung
.SV zähler=&zähler+1;.** Die Variable &zähler erhöhen
Durchlauf &zähler
.EW
&zähler war der letzte Durchlauf.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
Durchlauf 1
Durchlauf 2
Durchlauf 3
3 war der letzte Durchlauf.
```

Das Ergebnis der Bedingung muss sich von "wahr" in "falsch" ändern. Wenn das Ergebnis immer "wahr" wäre, würde Con-form die Schleife endlos wiederholen. Wenn die Bedingung schon beim ersten Mal das Ergebnis "falsch" liefert, wird die Schleife gar nicht durchlaufen, sondern sie wird übersprungen.



Anmerkung: Schleifen können nicht geschachtelt werden.

Eine Liste von Variablen mit ähnlichen Namen verarbeiten

Mit der Anweisung `.WH` können Sie mehrere Variablen verarbeiten, deren Namen ähnlich sind und mit einer laufenden Nummer enden. Definieren Sie dazu eine zusätzliche numerische Variable und erhöhen Sie deren Wert in der Schleife. Kombinieren Sie dann die Variablennamen mit der numerischen Variablen wie im folgenden Beispiel:

```
.SV name1=John
.SV name2=Paul
.SV name3=Ringo
.SV name4=George
.SV zahl=1;.** Dies ist die zusätzliche numerische Variable.
.WH &zahl <= 4
.** Den Anfang der Variablen &name mit der Variablen &zahl kombinieren:
&name&zahl
.SV zahl=&zahl+1;.** Die Variable &zahl erhöhen.
.EW
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
John
Paul
Ringo
George
```

.WX - Schleife verlassen

```
.WX
```

Mit dieser Anweisung wird eine Schleife verlassen, bevor ihr Ende erreicht ist. Die Verarbeitung fährt nach der Anweisung `.EW` fort. Im folgenden Beispiel beendet das Auftreten des Namens "Paul" die Ausführung der Schleife:

```
.SV name1=John
.SV name2=Paul
.SV name3=Ringo
.SV name4=George
.SV zahl=1
.WH &zahl <= 4
&name&zahl
.IF &name&zahl = Paul
.WX
.SV zahl=&zahl+1;
.EW
.SL 1
&name&zahl war der letzte Name in der Schleife.
```

Die formatierte Ausgabe sieht folgendermaßen aus:

```
John
Paul

Paul war der letzte Name in der Schleife.
```

.EW - Schleifenende

```
.EW
```

Diese Anweisung muss in Verbindung mit der Anweisung `.WH` benutzt werden. Sie definiert das Ende der Schleife.

Mit einer **.IF-Anweisung** ist es nicht möglich, die Verarbeitung von `.EW` bedingt zu unterdrücken.

Eine Bedingung definieren

Nach den Anweisungen `.IF` und `.WH` muss in derselben Zeile eine Bedingung stehen. Eine Bedingung besteht aus zwei Werten, die mit einem Vergleichsoperator verknüpft werden.

Wenn Sie nach der Anweisung `.IF` oder `.WH` mehrere Bedingungen definieren möchten, müssen Sie die einzelnen Bedingungen mit den logischen Operatoren `.AND` oder `.OR` verknüpfen (siehe [Logische Operatoren](#)).

Vergleichsoperatoren

Um eine Bedingung anzugeben, verknüpfen Sie zwei Werte mit einem der folgenden Vergleichsoperatoren:

Operator	Bedeutung
=	EQ gleich
<>	NE nicht gleich
<	LT kleiner als
<=	LE kleiner als oder gleich
>	GT größer als
>=	GE größer als oder gleich
?	Teilzeichenkette
*	Untermenge

Die meisten Vergleichsoperatoren können - wie in der Tabelle gezeigt - auf zwei verschiedene Arten angegeben werden. Um z.B. zwei Werte auf Gleichheit zu prüfen, können Sie entweder das Gleichheitszeichen (=) oder die Abkürzung EQ verwenden, d.h. die beiden folgenden Anweisungen sind identisch:

```
.IF &Alter = 50
.IF &Alter EQ 50
```

Vor und nach dem Vergleichsoperator muss jeweils mindestens ein Leerzeichen stehen.

Der erste Wert (links vom Vergleichsoperator) muss eine Variable sein.

Der zweite Wert (rechts vom Vergleichsoperator) kann eine Variable oder eine Konstante sein (z.B. eine Zahl oder ein Name).

In einigen Fällen ist es nicht nötig, den zweiten Wert anzugeben. Wenn Sie eine Konsequenz z.B. nur dann verarbeiten möchten, wenn eine Variable einen Wert hat (d.h. wenn der Wert nicht leer ist), können Sie angeben:

```
.IF &straße NE;.TH;.BR  
&straße  
.EI
```

Eine Konstante in Apostrophe einschließen

Wenn eine Konstante Leerzeichen oder Kommas enthält, muss sie in Apostrophe eingeschlossen werden, z.B.:

```
.IF &firma <> 'Software AG'
```

Wenn die Konstante selbst ein Apostroph enthält, muss es doppelt angegeben werden, z.B.:

```
.IF &frage <> 'Wie geht''s?'
```

Ein Parameter, der nicht in Apostrophe eingeschlossen werden muss, darf das Apostroph enthalten, z.B.:

```
.IF &text = geht's
```

Eine Konstante kann beliebig viele Apostrophe enthalten, solange das erste Zeichen kein Apostroph ist.

Teilzeichenkette (?)

Der Operator ? wird benutzt, um festzustellen, ob die links vom Operator angegebene Zeichenkette in der rechts vom Operator angegebenen enthalten ist.

Im folgenden Beispiel ist die linke Zeichenkette in der rechten enthalten. Das Ergebnis der Bedingung ist also wahr.

```
.IF ding ? Bedingungen
```

Im folgenden Beispiel ist die linke Zeichenkette *nicht* in der rechten enthalten. Das Ergebnis der Bedingung ist also falsch.

```
.IF dinge ? Bedingungen
```



Anmerkung: Der erste Wert (links vom Operator) muss nicht unbedingt eine Variable, sondern kann auch eine Konstante sein.

Untermenge (*)

Der Operator * wird benutzt, um festzustellen, ob jedes Zeichen, das in der linken Zeichenkette vorkommt, auch in der rechten Zeichenkette vorkommt.

In den folgenden Beispielen sind jeweils die Zeichen der linken Zeichenkette in der rechten Zeichenkette erhalten, d.h. die Ergebnisse dieser Bedingungen sind wahr.

```
.IF e * Blumentopf
.IF blut * Blumentopf
.IF ttt * Blumentopf
```

Im folgenden Beispiel sind *nicht* alle Zeichen der linken Zeichenkette in der rechten Zeichenkette enthalten. Das Ergebnis der Bedingung ist also falsch.

```
.IF abc * baden
```



Anmerkung: Der erste Wert (links vom Operator) muss nicht unbedingt eine Variable, sondern kann auch eine Konstante sein.

Eine Konsequenz definieren

Eine Konsequenz kann eine Anweisung und/oder Text sein. Sie wird verarbeitet, wenn das Ergebnis einer Bedingung wahr ist.

Wenn die Konsequenz eine Anweisung ist, können Sie sie in derselben Zeile angeben wie die Anweisung .IF oder .WH bzw. in derselben Zeile wie der letzte logische Operator (.AN oder .OR). Davor müssen Sie das Anweisungstrennzeichen eingeben (anfänglich das Semikolon). Beispiel:

```
.IF &Alter > 60;.TH
```

Wenn die Konsequenz aus Text besteht, muss er in einer neuen Zeile beginnen. Beispiel:

```
.IF &Alter = 50
Herzlichen Glückwunsch
```



Anmerkung: Die obigen Regeln gelten auch für eine Alternative, die mit der Anweisung .EL definiert wird.

Logische Operatoren

In den Anweisungen `.IF` und `.WH` können die logischen Operatoren `.AN` und `.OR` benutzt werden. Diese Operatoren werden wie getrennte Anweisungen behandelt. Sie müssen deshalb durch das Anweisungstrennzeichen (anfänglich das Semikolon) von der vorhergehenden Bedingung getrennt werden. Beispiel:

```
.IF &Alter = 50;.AN &Geschlecht = M
```

Wenn in derselben Zeile mehrere logische Operatoren stehen, werden sie von links nach rechts verarbeitet. Die Anweisung `.AN` oder `.OR` kann aber auch in eine neue Zeile geschrieben werden. Beispiel:

```
.IF &Alter = 50  
.AN &Geschlecht = M
```

.AN - Und

```
.AN bedingung
```

Wenn Sie den logischen Operator `.AN` verwenden, müssen alle Bedingungen wahr sein, damit die Konsequenz ausgeführt wird. Beispiel:

```
.SV Alter=65  
.SV Geschlecht=M  
.IF &Alter GE 65;.AN &Geschlecht = F  
Liebe Großmutter
```

Da in diesem Beispiel nur die erste Bedingung wahr ist, wird der Text "Liebe Großmutter" nicht ausgegeben.

.OR - Oder

```
.OR bedingung
```

Wenn Sie den logischen Operator `.OR` benutzen, muss mindestens eine der Bedingungen wahr sein, damit die Konsequenz ausgeführt wird. Beispiel:

```
.SV Alter=65  
.SV Geschlecht=M  
.IF &Alter GE 65;.OR &Geschlecht = F  
Liebe Großmutter
```

In diesem Beispiel ist nur die erste Bedingung wahr. Der Text "Liebe Großmutter" wird ausgegeben, da bei der Oder-Verknüpfung nur eine der Bedingungen zutreffen muss, um die Konsequenz auszuführen.

11 Makros

▪ Allgemeine Informationen	250
▪ Ein Makro definieren	251
▪ Ein Makro aufrufen	253
▪ Lokale Makrovariablen	254

Dieses Kapitel behandelt die folgenden Themen:

Allgemeine Informationen

Ein Makro ist eine Folge von Con-form-Anweisungen und/oder Text. Es ist sinnvoll, ein Makro zu definieren, wenn die gleichen Anweisungen oder der gleiche Text mehrfach in einem oder mehreren Dokumenten vorkommen.

Ein Makro kann beliebige Con-form-Anweisungen enthalten, mit Ausnahme einer zweiten `.MA-`Anweisung (d.h. Makros können nicht geschachtelt werden). Ein Makro kann andere Makros aufrufen. Rekursive Makros sind nicht möglich, d.h. ein Makro darf sich nicht selbst aufrufen (weder direkt noch über ein anderes Makro).

Alle Makros, die Sie definieren, sollten in einem extra Dokument enthalten sein. Dieses Dokument, das so genannte "Formatierprofil", wird immer vor dem Rohdokument verarbeitet (siehe [Formatierprofile](#)).

Wenn Sie ein Makro nicht im Formatierprofil definieren, können Sie das Makro nur in dem Dokument benutzen, in dem es definiert wurde. In diesem Fall wird der Makroaufruf nur als gültige Anweisung erkannt, wenn er nach der Makrodefinition auftritt.

Eine weitere Möglichkeit, ein Makro in Ihrem Büro allgemein verfügbar zu machen, ist folgende: Legen Sie für jede Makrodefinition ein extra Dokument an. Im Rohtext müssen Sie dann die folgenden Anweisungen angeben, um das Makro aufzurufen:

```
.EM dokumentname  
.makroname
```

Geben Sie als Parameter *dokumentname* den Namen des Dokuments an, das die Makrodefinition enthält (siehe die Anweisung `.EM`). Geben Sie als Parameter *makroname* den Namen des Makros an, das aufgerufen werden soll.

Ein Makro definieren

.MA - Beginn eines Makros

```
.MA makroname
```

Die Anweisung `.MA` zeigt an, dass eine Makrodefinition beginnt. Als Parameter muss der Name des Makros in derselben Zeile eingegeben werden. Der Name kann bis zu 100 Zeichen haben; zwischen Groß- und Kleinbuchstaben wird nicht unterschieden.

Um z.B. ein Makro namens "Vorgaben" zu definieren, geben Sie an:

```
.MA Vorgaben
```



Tipp: Es wird empfohlen, Makronamen mit mindestens 3 Zeichen zu definieren, damit sie nicht mit Con-form-Anweisungen verwechselt werden können.

Geben Sie unter der Anweisung `.MA` die Makrodefinition ein (d.h. die Anweisungen und/oder Textzeilen, die das Makro enthalten soll). Es folgt ein Beispiel eines Makros, das Vorgabewerte definiert:

```
.MA Vorgaben
.LM 10;.RM 60
.FI ON;.JU ON
.OP ESC=/
.OP HYP=E
.IF &$LC < 5:.NP
.ME
```

Die Makrodefinition wird mit der Anweisung `.ME` abgeschlossen.

Eine Con-form-Anweisung deaktivieren

Wenn Sie den Namen einer Con-form-Anweisung als Makronamen benutzen, kann die Con-form-Anweisung nicht mehr ausgeführt werden. Jedes Mal, wenn der Name im Rohdokument vorkommt, wird anstelle der Con-form-Anweisung das Makro ausgeführt.

Auf diese Weise können Sie eine Con-form-Anweisung deaktivieren. Das unten gezeigte Makro z.B. hat keine Wirkung, da weder Text noch Anweisungen für das Makro definiert wurden. Es macht jedoch die Con-form-Anweisung `.NP`, die normalerweise einen Seitenvorschub bewirkt, unwirksam. Dies ist sinnvoll, wenn Sie z.B. ein Dokument zum Korrekturlesen drucken möchten, wobei es wichtiger ist, Papier zu sparen anstatt das endgültige Seiten-Layout zu erhalten.

.MA NP
.ME

.MX - Makro abbrechen

```
.MX
```

Sie können die Anweisung `.MX` an einer beliebigen Stelle innerhalb des Makros angeben, um das Makro zu verlassen, bevor sein Ende erreicht ist. Die Verarbeitung fährt dann nach der Anweisung `.ME` fort. Die nach der Anweisung `.MX` stehenden Makrodefinitionen werden nicht verarbeitet.

Wenn Sie die Anweisung `.MX` in einer **.IF-Anweisung** verwenden, wird empfohlen, sie nur in einer einfachen Konsequenz anzugeben, die keine `.EI`-Anweisung erfordert.



Anmerkung: Wenn Sie die Anweisung `.MX` in einer Blockkonsequenz (`.TH`) oder Alternative (`.EL`) benutzen, wird das Makro verlassen, bevor die Anweisung `.EI` auftritt. Da Con-form in diesem Fall nicht erkennt, dass Sie die Anweisung `.EI` angegeben haben, erscheint eine Fehlermeldung.

.ME - Ende eines Makros

```
.ME
```

Diese Anweisung schließt eine Makrodefinition ab. Optional können Sie den Namen des Makros angeben, z.B.:

```
.ME Vorgaben
```

Mit einer **.IF-Anweisung** ist es nicht möglich, die Verarbeitung von `.ME` bedingt zu unterdrücken.

Ein Makro aufrufen

Um ein Makro aufzurufen, geben Sie im Rohtext einen Punkt (`.`) gefolgt vom Makronamen an. Wenn Sie z.B. das Makro "Vorgaben" aufrufen möchten, geben Sie an:

```
.Vorgaben
```

Daraufhin werden die Anweisungen und/oder Textzeilen verarbeitet, die für das Makro definiert wurden. Die Verarbeitung beginnt direkt nach der Anweisung `.MA` und endet entweder mit der Anweisung `.MX` oder der Anweisung `.ME`, die das Makroende anzeigt.

Lokale Makrovariablen

Wenn Sie lokale Makrovariablen definieren, können Sie einem Makro Parameter übergeben (z.B. können Sie Name und Adresse an ein Makro übergeben, das einen Standardbrief erzeugt).

Außer den lokalen Makrovariablen können Sie in der Makrodefinition oder als Parameter im Makroaufruf auch **Systemvariablen** benutzen oder Variablen, die Sie mit der Anweisung **.SV** definierten. Wenn Sie Makros definieren, die die Werte dieser Variablen ändern, ist besondere Vorsicht geboten, da leicht Verwirrung entstehen kann, wenn verschiedene Makros die gleiche Variable ändern.

Die Ersetzung von Variablen (siehe die Anweisung **.SU**) wird von einem Makroaufruf *mit* Parametern automatisch eingeschaltet. Ein Makroaufruf ohne Parameter schaltet die Ersetzung nicht ein.

Makrodefinition mit Parametern

Eine Makrodefinition kann bis zu 99 lokale Makrovariablen enthalten. Die Namen der lokalen Makrovariablen lauten: \$1, \$2, \$3, ... \$99. Vor jeder Makrovariablen in der Makrodefinition muss das Variablenzeichen (anfänglich das kaufmännische Und (&)) angegeben werden.

Später beim Makroaufruf müssen Sie die Parameter (d.h. die entsprechenden Werte für die Variablen) nach dem Makronamen angeben (siehe *Makroaufruf mit Parametern*).

In der folgenden Makrodefinition bestimmt die Variable \$1 die Anzahl der Leerzeilen, die im Dokumenttext eingefügt werden sollen, und die Variable \$2 den Text, der nach den Leerzeilen ausgegeben werden soll:

```
.MA abbildung
.IL &$1
.CE 1
&$2
.SL 2
.ME
```

Die Variable \$1 in der Makrodefinition muss beim Makroaufruf der erste Parameter sein. Die Variable \$2 muss als zweiter Parameter angegeben werden. Beispiel für den Aufruf dieses Makros mit zwei Parametern:

```
.abbildung 5 Statistik
```

Als Ergebnis werden 5 Leerzeilen (Variable \$1) in den Text eingefügt, und der Text "Statistik" (Variable \$2) wird nach den Leerzeilen zentriert ausgegeben.

Die Anzahl der Parameter im Makroaufruf bestimmen

Die lokale Makrovariable \$0 enthält die Anzahl der Parameter, die beim Makroaufruf nach dem Makronamen angegeben wurden. Beispiel:

```
.abbildung 5 Statistik
Anzahl der Parameter im aktuellen Makroaufruf: &$0
```

Im obigen Beispiel wurden beim Makroaufruf zwei Parameter angegeben, d.h. die Variable \$0 erhält den Wert 2. Die formatierte Ausgabe sieht folgendermaßen aus:

```
Anzahl der Parameter im aktuellen Makroaufruf: 2
```

Wenn Sie ein Makro ohne Parameter aufrufen, wird der lokalen Makrovariablen \$0 der Wert 0 (null) zugewiesen.

Beim Makroaufruf die richtige Anzahl von Parametern angeben

Sie können die lokale Makrovariable \$0 benutzen, um sicherzustellen, dass die richtige Anzahl von Parametern an ein Makro übergeben wird. Beispiel:

```
.MA abbildung
.IF &$0 NE 2
.TH
*****
Dieses Makro hat 2 Parameter:
Parameter 1: Anzahl der Leerzeilen für die Abbildung
Parameter 2: Überschrift der Abbildung
*****
.EL
.IL &$1
.CE 1
&$2
.EI
.ME
```

In diesem Beispiel wird die Anweisung .IF benutzt, um sicherzustellen, dass beim Makroaufruf genau zwei Parameter angegeben werden. Sind es mehr oder weniger Parameter, wird eine entsprechende Meldung ausgegeben.

Makroaufruf mit Parametern

Wenn ein Makro Parameter hat, müssen Sie die entsprechenden Werte beim Makroaufruf nach dem Makronamen angeben. Dabei muss zwischen Makroname und Parameter genau ein Leerzeichen stehen. Beispiel:

```
.kap1 Überblick
```

Parameter in Apostrophe einschließen

Wenn ein Parameter Leerzeichen oder Kommas enthält, müssen Sie ihn in Apostrophe einschließen. Beispiel:

```
.kap1 'Überblick über Con-form'
```

Wäre dieser Parameter nicht in Apostrophe eingeschlossen, würde Con-form annehmen, es handle sich bei den drei Wörtern um drei Parameter.



Anmerkung: Makroparameter, die keine Leerzeichen oder Kommas enthalten, können in Apostrophe eingeschlossen werden; es ist jedoch nicht erforderlich.

Wenn ein Parameter, der in Apostrophe eingeschlossen werden muss, selbst einen Apostroph enthält, müssen Sie den Apostroph doppelt angeben. Beispiel:

```
.kap1 'Wie geht''s weiter?'
```

Ein nicht in Apostrophen eingeschlossener Parameter kann selbst beliebig viele Apostrophe enthalten, jedoch darf das erste Zeichen kein Apostroph sein.

Makroaufruf mit mehreren Parametern

Wenn ein Makro mehrere Parameter hat, muss zwischen den Parametern jeweils ein Komma und/oder Leerzeichen stehen. Das folgende Beispiel hat zwei Parameter, getrennt durch Komma und Leerzeichen (es ist auch möglich, sie nur durch ein Leerzeichen zu trennen):

```
.bsp 15, 'Ein Makro definieren'
```

Jeder Parameter, den Sie im Makroaufruf angeben, wird einer lokalen Makrovariablen zugewiesen. Im obigen Beispiel erhält die Variable \$1 den Wert 15, und die Zeichenkette "Ein Makro definieren" wird der Variablen \$2 zugewiesen.

Die lokalen Makrovariablen sind nur innerhalb des aufgerufenen Makros bekannt. Sobald die Ausführung des Makros beendet ist, sind die lokalen Makrovariablen nicht mehr verfügbar.

12 Steueranweisungen für die Ausgabe

▪ Die auszugebenden Seiten definieren	260
▪ Die Papierkassette definieren	262
▪ Leerzeichen	264
▪ Zeichen umsetzen	267
▪ Die Textrichtung ändern	276
▪ Fehlerbehebung	278

Der Begriff "Ausgabe" bezieht sich auf die formatierte Version eines Dokuments, die entweder gedruckt, am Bildschirm angezeigt oder als neues Dokument abgelegt werden kann.

Dieses Kapitel behandelt die folgenden Themen:

Die auszugebenden Seiten definieren

.OP NPG - Anzahl der auszugebenden Seiten

```
.OP NPG=zah1
```

Sie können angeben, wie viele formatierte Seiten Sie ausgeben möchten. Um z.B. die ersten 5 Seiten eines Dokuments auszugeben, geben Sie am Anfang des Rohtextes die folgende Anweisung an:

```
.OP NPG=5
```

Index und Inhaltsverzeichnis werden immer nach dem Dokumenttext ausgegeben. Sie werden von dieser Anweisung nicht beeinflusst.

.OP STA - Ausgabe ab einer bestimmten Seite

```
.OP STA=zah1
```

Sie können die Ausgabe des formatierten Dokuments mit einer bestimmten Seite beginnen. Alle Seiten vor der angegebenen Seitenzahl werden ignoriert. Um die Ausgabe z.B. mit Seite 11 zu beginnen, geben Sie die folgende Anweisung am Anfang des Rohtextes ein:

```
.OP STA=11
```



Anmerkung: Wenn Sie die obige Anweisung z.B. auf Seite 5 angeben, werden alle Seiten bis zu dieser Anweisung auf Seite 5 ebenfalls ausgegeben.

Index und Inhaltsverzeichnis werden (falls Einträge dafür vorhanden sind) nur für die ausgegebenen Seiten erstellt.

.OP STO - Ausgabe bis zu einer bestimmten Seite

```
.OP STO=zah1
```

Sie können die Ausgabe des formatierten Dokuments an einer bestimmten Seite beenden. Alle Seiten nach der angegebenen Seitenzahl werden ignoriert. Um die Ausgabe z.B. mit Seite 20 zu beenden, geben Sie die folgende Anweisung am Anfang des Rohtextes (irgendwo vor Seite 20) an:

```
.OP STO=20
```

Index und Inhaltsverzeichnis werden (falls Einträge dafür vorhanden sind) nur für die ausgegebenen Seiten erstellt.

Die Papierkassette definieren

Wenn Ihr Drucker zwei Papierkassetten hat, können Sie angeben, welche benutzt werden soll. Die Anweisungen `.OP BIN` und `.OP SSF` müssen in Verbindung mit einem Druckerprofil verwendet werden.

.OP BIN - Papierkassette wählen

```
.OP BIN=1  
.OP BIN=2
```

Sie können eine von zwei Papierkassetten auswählen.

Um Kassette 1 auszuwählen, geben Sie an:

```
.OP BIN=1
```

Um Kassette 2 auszuwählen, geben Sie an:

```
.OP BIN=2
```

Wenn die Anweisung `.OP BIN` im Rohtext vorkommt, erstellt Con-form intern die folgenden Symbole:

F1	Seitenvorschub Kassette 1 (bei <code>.OP BIN=1</code>)
F2	Seitenvorschub Kassette 2 (bei <code>.OP BIN=2</code>)

Wenn Sie keine Papierkassette angeben, erstellt Con-form intern das folgende Symbol:

FF	Seitenvorschub
----	----------------

Für jedes Symbol müssen Sie die entsprechende Druckerbefehlsfolge (die in Ihrem Druckerhandbuch dokumentiert ist) in Ihrem Druckerprofil definieren (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Druckerprofile*).

.OP SSF - Papierkassette wechseln

```
.OP SSF=ON  
.OP SSF=OFF
```

Wenn die erste Seite der Ausgabe aus Papierkassette 1 und die folgenden Seiten aus Papierkassette 2 genommen werden sollen, geben Sie an:

```
.OP SSF=ON
```

Dies ist sinnvoll, wenn Sie die erste Seite eines Dokuments auf anderem Papier, z.B. auf dickerem oder farbigem Papier, ausdrucken möchten.



Anmerkung: Wenn .OP SSF=ON angegeben ist, werden damit unvereinbare .OP BIN-Einstellungen ignoriert.

Anfänglich wird das Papier immer der gleichen Papierkassette entnommen. Dies entspricht der Anweisung:

```
.OP SSF=OFF
```

Leerzeichen

.CB - Leerzeichen unterdrücken

```
.CB ON  
.CB OFF
```

Sie können mehrfache Leerzeichen in Ihrem Rohtext durch einzelne Leerzeichen ersetzen. Dazu muss der Zeilenausgleich eingeschaltet sein (.FI ON).

Um mehrfache Leerzeichen durch einzelne Leerzeichen zu ersetzen, geben Sie an:

```
.CB ON
```

Wenn sowohl der Zeilenausgleich als auch der Randausgleich eingeschaltet sind (.FI ON;.JU ON), können wegen des Blocksatzes einzelne Leerzeichen anschließend wieder "gedehnt" werden (d.h. zwischen Wörtern werden zusätzliche Leerzeichen eingefügt).

Die anfängliche Einstellung dieser Anweisung ist:

```
.CB OFF
```

.SB - Leerzeichen am Zeilenanfang ignorieren

```
.SB ON  
.SB OFF
```

Ein oder mehrere Leerzeichen am Anfang einer Zeile verursachen einen Zeilenumbruch. Um die Leerzeichen am Zeilenanfang zu ignorieren, geben Sie an:

```
.SB ON
```

Die anfängliche Einstellung dieser Anweisung ist:

`.SB OFF`

Sie können die Anweisung `.SB` (anders als die Anweisung `.CB`) auch dann verwenden, wenn der Zeilenausgleich ausgeschaltet ist (`.FI OFF`). In diesem Fall beginnt die Zeile am linken Rand - ohne die vorausgehenden Leerzeichen.

Sie können auch "feste Leerzeichen" definieren (siehe die Anweisung `.TR`).

Beispiel

Dieses Beispiel zeigt, wie die Anweisungen `.CB` und `.SB` verwendet werden.

Rohtext

```
.LM 0;.RM 65
.FI ON;.JU OFF
.CB ON
Sie können alle mehrfachen Leerzeichen durch einzelne
Leerzeichen ersetzen.
.IL 1
.FI ON;.JU ON
Wenn der Zeilenausgleich eingeschaltet ist, können
einzelne Leerzeichen jedoch wieder "gedehnt" werden,
damit der Blocksatz erreicht wird.
.CB OFF
.IL 1
Ein Leerzeichen am Zeilenanfang bewirkt einen
Zeilenumbruch. Das Leerzeichen erscheint jedoch in der
formatierten Ausgabe am Anfang der Zeile.
.IL 1
.SB ON
Wenn die Anweisung .SB eingeschaltet ist, wird ein Leerzeichen
am Zeilenanfang ignoriert.
.IL 1
.FI OFF
Sie können die Anweisung .SB (im Gegensatz zur Anweisung .CB)
auch benutzen, wenn der Zeilenausgleich ausgeschaltet ist.
```

Formatierte Ausgabe

Sie können alle mehrfachen Leerzeichen durch einzelne Leerzeichen ersetzen.

Wenn der Zeilenausgleich eingeschaltet ist, können einzelne Leerzeichen jedoch wieder "gedehnt" werden, damit der Blocksatz erreicht wird.

Ein Leerzeichen am Zeilenanfang bewirkt einen Zeilenumbruch. Das Leerzeichen erscheint jedoch in der formatierten Ausgabe am Anfang der Zeile.

Wenn die Anweisung `.SB` eingeschaltet ist, wird ein Leerzeichen am Zeilenanfang ignoriert.

Sie können die Anweisung `.SB` (im Gegensatz zur Anweisung `.CB`) auch benutzen, wenn der Zeilenausgleich ausgeschaltet ist.

Zeichen umsetzen

Sie können ein bestimmtes Zeichen aus dem Rohtext in der formatierten Ausgabe durch ein anderes Zeichen oder eine Zeichenkette ersetzen.

.TR - Zeichen umsetzen

```
.TR eingabezeichen ausgabezeichen
.TR
```

Mit dieser Anweisung geben Sie für ein bestimmtes Zeichen des Rohtextes ein anderes Zeichen in der formatierten Ausgabe aus. Die Parameter sind jeweils ein einzelnes Zeichen oder deren entsprechende hexadezimale Darstellung.

Der erste Parameter (Eingabezeichen) definiert das Zeichen, das Sie umsetzen möchten; der zweite Parameter (Ausgabezeichen) definiert das Zeichen, das stattdessen ausgegeben werden soll. Zwischen den beiden Parametern muss ein Leerzeichen stehen.

Sie können so viele verschiedene .TR-Anweisungen angeben, wie erforderlich sind.

Ein einzelnes Zeichen umsetzen

Um z.B. den Kleinbuchstaben "a" in den Großbuchstaben "A" umzusetzen, geben Sie an:

```
.TR a A
```

Wenn Sie das Anweisungstrennzeichen (anfänglich das Semikolon) als Parameter angeben möchten, müssen Sie es doppelt eingeben (;:).

Hexadezimale Darstellung

Wenn Sie auf einem Mainframe arbeiten und den Kleinbuchstaben "a" in den Großbuchstaben "A" umsetzen möchten, müssen Sie die Anweisung .TR mit den entsprechenden EBCDIC-Zeichen angeben:

```
.TR 81 C1
```

Wenn Con-form auf einem Mainframe installiert ist, sollten die folgenden Zeichen nicht in der Anweisung .TR benutzt werden (X ist das umzusetzende Zeichen):

```
.TR X 00
.TR X 03
```

.TR X 04
.TR X 0E
.TR X 0F
.TR X 27
.TR X 3F
.TR X 40
.TR X FE

Ein "festes Leerzeichen" angeben

Wenn Sie sicherstellen möchten, dass beim Zeilenausgleich keine zusätzlichen Leerzeichen zwischen zwei Wörtern eingefügt werden, können Sie im Rohtext an den entsprechenden Stellen feste Leerzeichen definieren. An diesen Stellen findet auch kein Zeilenumbruch statt.

Um ein festes Leerzeichen zu definieren, müssen Sie ein hexadezimaleres Ausgabezeichen angeben.

40 ist die EBCDIC-Darstellung für das feste Leerzeichen. Beispiel:

```
.TR ^ 40
Siehe Kapitel^7.
```

In diesem Beispiel wurde das definierte Eingabezeichen (^) zwischen dem Wort "Kapitel" und der Kapitelnummer "7" eingefügt. In der formatierten Ausgabe werden diese in derselben Zeile mit genau einem Leerzeichen dazwischen ausgegeben.

Eine Zeichenumsetzung rückgängig machen

Um das Umsetzen eines Zeichens wieder rückgängig zu machen, müssen Sie das entsprechende Zeichen in beiden Parametern der Anweisung .TR angeben. Um z.B. die Anweisung .TR a A rückgängig zu machen, geben Sie an:

```
.TR a a
```

Alle Zeichenumsetzungen rückgängig machen

Um alle Zeichenumsetzungen wieder rückgängig zu machen, geben Sie die Anweisung .TR ohne Parameter an:

```
.TR
```

Alle Umsetzungen werden ausgeführt, *nachdem* eine Rohzeile formatiert wurde. Deshalb ist die Position der Anweisung .TR (die alle Zeichenumsetzungen rückgängig machen soll) wichtig, besonders bei eingeschaltetem Zeilenausgleich (.FI ON). Dies zeigt das folgende Beispiel:

```
.LM 0;.RM 72
.FI ON;.JU OFF
.TR a A
Jedes kleine "a" wird in ein großes "A" umgewandelt.
.TR
Die formatierte Ausgabe zeigt jedoch ein unerwartetes Ergebnis,
da die Anweisung .TR (die alle Umsetzungen rückgängig macht) an
einer Stelle im Rohtext steht, wo die gesamte Zeile noch nicht
```

formatiert (d.h. durch den Zeilenausgleich gefüllt) worden ist.

.IL

.TR a A

Wenn Sie ein bestimmtes Zeichen in einem Abschnitt durch ein anderes Zeichen ersetzen möchten, sollte die Anweisung .TR (die alle Umsetzungen rückgängig macht) nach dem nächsten Zeilenumbruch angegeben werden.

.IL;.TR

Die formatierte Ausgabe sieht folgendermaßen aus:

Jedes kleine "a" wird in ein großes "A" umgewandelt. Die formatierte Ausgabe zeigt jedoch ein unerwartetes Ergebnis, da die Anweisung .TR (die alle Umsetzungen rückgängig macht) an einer Stelle im Rohtext steht, wo die gesamte Zeile noch nicht formatiert (d.h. durch den Zeilenausgleich gefüllt) worden ist.

Wenn Sie ein bestimmtes Zeichen in einem Abschnitt durch ein Anderes Zeichen ersetzen möchten, sollte die Anweisung .TR (die Alle Umsetzungen rückgängig macht) nach dem nächsten Zeilenumbruch angegeben werden.

.TS - Ein Zeichen in eine Zeichenkette umsetzen

```
.TS eingabezeichen ausgabezeichenkette  
.TS eingabezeichen  
.TS
```

Mit dieser Anweisung setzen Sie ein bestimmtes Zeichen in eine Zeichenkette um. Der erste Parameter (Eingabezeichen) definiert das umzusetzende Zeichen; der zweite Parameter (Ausgabezeichenkette) gibt an, wie das Zeichen ausgegeben werden soll. Zwischen beiden Parametern muss ein Leerzeichen stehen.

Wie bei der Anweisung .TR muss als Eingabezeichen entweder ein einzelnes Zeichen oder seine entsprechende hexadezimale Darstellung angegeben werden. Die Ausgabezeichenkette kann aus mehreren Zeichen bestehen. Sie darf Leerzeichen enthalten, jedoch *keine* hexadezimalen Werte.

Sie können so viele verschiedene .TS-Anweisungen angeben, wie erforderlich sind.



Tip: Wenn Sie ein Zeichen in ein anderes einzelnes Zeichen umsetzen möchten, wird empfohlen, die Anweisung .TR zu verwenden.

Um z.B. das Zeichen "/" in die Zeichenkette "Schrägstrich" umzusetzen, geben Sie an:

```
.TS / Schrägstrich
```

Wenn das Eingabezeichen in der Ausgabezeichenkette wieder auftritt, wird es nicht noch einmal umgesetzt. Wenn Sie das Anweisungstrennzeichen (anfänglich das Semikolon) als Parameter angeben möchten, müssen Sie es doppelt eingeben (;;).

Alle Umsetzungen werden - anders als bei der Anweisung .TR - ausgeführt, *bevor* eine Rohertextzeile formatiert wird.



Anmerkung: Alle Zeichen im Rohertext werden verarbeitet, auch diejenigen, die das Ergebnis einer Variablenersetzung sind (siehe die Anweisung [.SV](#)).

Eine Zeichenumsetzung rückgängig machen

Um das Umsetzen eines Zeichens wieder rückgängig zu machen, müssen Sie das Zeichen angeben, das Sie vorher als Eingabezeichen definierten. Die Ausgabezeichenkette müssen Sie nicht angeben.

Um z.B. die Umsetzung rückgängig zu machen, die durch die Anweisung .TS / Schrägstrich (siehe oben) definiert wurde, geben Sie an:

```
.TS /
```

Alle Zeichenumsetzungen rückgängig machen

Um alle Zeichenumsetzungen wieder rückgängig zu machen, geben Sie die Anweisung .TS ohne Parameter an:

```
.TS
```

Beispiel

Dieses Beispiel zeigt die Anweisungen .TR und .TS.

Rohtext

```
.TR - =
Der Bindestrich wird durch das Gleichheitszeichen ersetzt.
-----
.TR - -;.** Die vorherige Umsetzung rückgängig machen.
Nun wird der Bindestrich (-) in der formatierten Ausgabe nicht
umgesetzt.
.LM 0;.RM 45
.FI ON;.JU ON
.IL 1
.** Den Unterstrich als festes Leerzeichen benutzen.
.** Dieses Beispiel funktioniert in einer Mainframe-Umgebung.
.** Die hexadezimale Darstellung für das Leerzeichen ist 40.
.TR _ 40
Wenn Sie im Rohtext_feste_Leerzeichen definieren,_werden in_der
formatierten Ausgabe_an_diesen_Stellen keine zusätzlichen
Leerzeichen eingefügt. Außerdem erfolgt an der_Position_eines_
festen_Leerzeichens kein Zeilenumbruch.
.TR;.IL 1
.LM 0;.RM 70
.TS * Dieser Ausgabebetext wird vom Stern (*) verursacht.
.TS / Noch ein Ausgabebetext, diesmal verursacht vom Schrägstrich (/).
Wenn ein Stern im Rohtext vorkommt, wird folgendes ausgegeben: *
Wenn ein Schrägstrich im Rohtext vorkommt, wird folgendes
ausgegeben: /
.IL 1
.TS;.** Alle Umsetzungen rückgängig machen.
Wie im obigen Beispiel gezeigt, können Sie das Eingabezeichen (* und /)
in den Ausgabebetext einfügen. In diesem Fall wird das Zeichen nicht
noch einmal umgewandelt.
```

Formatierte Ausgabe

```
Der Bindestrich wird durch das Gleichheitszeichen ersetzt.
=====
Nun wird der Bindestrich (-) in der formatierten Ausgabe nicht
umgesetzt.

Wenn Sie im Rohtext feste Leerzeichen
definieren, werden in der formatierten
Ausgabe an diesen Stellen keine zusätzlichen
Leerzeichen eingefügt. Außerdem erfolgt an
der Position eines festen Leerzeichens kein
Zeilenumbruch.
```

Wenn ein Stern im Rohtext vorkommt, wird folgendes ausgegeben: Dieser Ausgabertext wird vom Stern (*) verursacht. Wenn ein Schrägstrich im Rohtext vorkommt, wird folgendes ausgegeben: Noch ein Ausgabertext, diesmal verursacht vom Schrägstrich (/).

Wie im obigen Beispiel gezeigt, können Sie das Eingabezeichen (* und /) in den Ausgabertext einfügen. In diesem Fall wird das Zeichen nicht noch einmal umgewandelt.

Die Textrichtung ändern

.TO - Textrichtung ändern

```
.TO L  
.TO R
```

Diese Anweisung ist sinnvoll, wenn Sie Text in Ihr Dokument aufnehmen möchten, der während des Formatiervorgangs von rechts nach links verarbeitet werden soll (z.B. wenn Sie ein hebräisches Zitat in den Text einfügen möchten).

Diese Anweisung hat keine Auswirkung auf die Darstellung oder die Zeilenformatierung, da Con-form annimmt, dass der betreffende Text von der entsprechenden Hardware verarbeitet wird. Sie übergibt lediglich den Wert für die geltende Textrichtung an die Routine, die benutzt wird, um RFT-DCA-Dokumente in Con-form-Dokumente (und umgekehrt) zu konvertieren.

Anfänglich wird der Text von links nach rechts verarbeitet. Dies entspricht der Anweisung:

```
.TO L
```

Um den Text von rechts nach links zu verarbeiten, geben Sie an:

```
.TO R
```

Die Bedeutung von "links" und "rechts" für die Anweisungen .LM und .RM muss nun anders interpretiert werden. Der linke Rand ist z.B. die Seite, wo Sie zu schreiben beginnen (das ist nun physisch die rechte Seite), und der rechte Rand ist die Seite, wo Sie aufhören zu schreiben (physisch die linke Seite).

Auch wenn Text von rechts nach links verarbeitet wird, werden die arabischen Seitenzahlen (definiert durch das Seitenzahlzeichen - anfänglich das Zeichen #) immer von links nach rechts verarbeitet.



Anmerkung: Siehe auch die [Textfunktion I](#). Diese Textfunktion bewirkt, dass der Inhalt einer Variablen in umgekehrter Richtung verarbeitet wird.

Die Textrichtung mit Umschaltanweisungen ändern

Sie können auch Umschaltanweisungen verwenden, um die Richtung zu ändern, in der Ihr Text verarbeitet wird. Dazu müssen Sie zuerst das Umschaltzeichen definieren (siehe [.OP ESC - Umschaltzeichen definieren](#)).

Wenn Ihr Text von links nach rechts verarbeitet wird, können Sie die folgenden Symbole in den Rohtext eingeben, um die Richtung für den markierten Text zu ändern:

Z1	Richtung von-rechts-nach-links einschalten
Z0	Richtung von-rechts-nach-links ausschalten

Wenn Ihr Text von rechts nach links verarbeitet wird, können Sie die folgenden Symbole in den Rohtext eingeben, um die Richtung für den markierten Text zu ändern:

A1	Richtung von-links-nach-rechts einschalten
A0	Richtung von-links-nach-rechts ausschalten

Die obigen Symbole werden während der Zeilenformatierung verarbeitet. Sie können sie auch mit anderen Symbolen kombinieren (z.B. M1 und M0). Dabei müssen Sie darauf achten, die Symbole korrekt zu schachteln.

Für Text, der die obigen Symbole enthält, dürfen Sie das Tabulatorzeichen (siehe [.TB - Tabulator setzen](#)) nicht verwenden.

Fehlerbehebung

.OP ECH - Anweisungen ausgeben

```
.OP ECH=ON  
.OP ECH=OFF
```

Sie können alle im Rohtext enthaltenen Con-form-Anweisungen in der formatierten Ausgabe vor dem jeweiligen Text ausgeben. Dies ist hilfreich, wenn die Ausgabe nicht wie gewünscht formatiert wurde und Sie überprüfen wollen, ob die Con-form-Anweisungen im Rohtext korrekt angegeben wurden.

Um die Con-form-Anweisungen auszugeben, geben Sie an:

```
.OP ECH=ON
```

Diese Option ist anfänglich ausgeschaltet. Dies entspricht der Anweisung:

```
.OP ECH=OFF
```

Wenn Sie nur einen Teil des Rohdokuments überprüfen möchten, können Sie den betreffenden Text zwischen die beiden obigen Anweisungen einschließen.



Anmerkung: Die Con-form-Anweisungen werden nicht in allen Umgebungen ausgegeben.

.DU - Speicherauszug drucken

```
.DU
```

Diese Anweisung ist eine Diagnostikhilfe und sollte nur auf Anforderung des Software AG Support benutzt werden. Geben Sie die Anweisung .DU im Rohtext direkt vor dem Punkt an, an dem das Programm mit Fehler endete, und drucken Sie dann die formatierte Version des Rohtextes aus.

13

Ein Dokument formatieren

- Der Befehl FORMATIEREN 280
- Druckerprofile 280
- Formatierprofile 281

Wenn Sie ein Dokument in Con-nect anlegen, ist Cnf das Format für ein Dokument, das Con-form-Anweisungen enthält.

Dieses Kapitel behandelt die folgenden Themen:

Der Befehl FORMATIEREN

Um die Con-form-Anweisungen, die im Dokumenttext enthalten sind, auszuführen, geben Sie den Befehl FORMATIEREN zusammen mit dem Dokument ein. Daraufhin erscheint der "Formatieren *Dokumentname*"-Bildschirm, in dem Sie angeben können, ob die formatierte Ausgabe angezeigt, gedruckt oder abgelegt werden soll. Sie können auch angeben, dass die formatierte Ausgabe in den Con-nect-Editor gestellt wird, damit Sie sie ändern können. Der Befehl FORMATIEREN ist im *Con-nect Benutzerhandbuch* im Abschnitt *Dokumente* beschrieben.

Wenn Sie die formatierte Ausgabe am Bildschirm anzeigen, ist diese Anzeige abhängig von der Umgebung, in der Sie arbeiten. In einer Mainframe-Umgebung z.B. wird hervorgehobener Text mit höherer Intensität angezeigt, aber Sie können den Formattyp (z.B. fett oder unterstrichen) nicht unterscheiden. Wenn Sie das formatierte Dokument drucken, werden die Hervorhebungen korrekt ausgegeben - vorausgesetzt, Sie benutzen das richtige Druckerprofil.

Sie können ein RFT-Dokument in das Format Cnf UEBERSETZEN (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Dokumente übersetzen*).

Druckerprofile

Bestimmte Funktionen von Con-form erfordern ein Druckerprofil. Druckerprofile sind abhängig von der Umgebung, in der Sie arbeiten. Wie ein Druckerprofil definiert wird, ist im *Con-nect Benutzerhandbuch* im Abschnitt *Druckerprofile* beschrieben.

Con-form unterstützt keine Proportionalschriftarten.

Der Druckername und der Name des Druckerprofils werden in Ihrem Con-nect-Teilnehmerprofil definiert (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Druckvorgaben*).

Wenn Sie kein Druckerprofil verwenden, oder wenn Ihr Druckerprofil falsche Angaben enthält oder Angaben fehlen, kann die gedruckte Ausgabe unerwartete Ergebnisse liefern.

Formatierprofile

Sie können ein Formatierprofil definieren, das jeweils vor dem Rohdokument verarbeitet wird. Das Formatierprofil kann z.B. Vorgabewerte für das Seiten-Layout enthalten. Auch Makros sollten im Formatierprofil stehen, damit sie immer verfügbar sind (andernfalls sind sie nur in dem Dokument gültig, in dem sie definiert wurden).

Für das Formatierprofil wird automatisch der Name FPROFILE in Ihrem Teilnehmerprofil definiert (siehe *Con-nect Benutzerhandbuch*, Abschnitt *Druckvorgaben*). Ein Dokument namens FPROFILE wird jedoch *nicht* mit Con-nect ausgeliefert. Es wird empfohlen, dass Ihr Administrator ein Dokument mit diesem Namen im Büro SYSCNT anlegt und damit für alle Büros zur Verfügung stellt.

Sie können auch ein anderes Formatierprofil (d.h. den Namen eines anderen Dokuments) in Ihrem Teilnehmerprofil angeben. In diesem Fall verarbeitet Con-form zuerst das Systemformatierprofil im Büro SYSCNT und dann Ihr eigenes Formatierprofil. Dies ist sinnvoll, wenn Sie z.B. ein Makro aus dem Systemformatierprofil durch ein erweitertes eigenes Makro ersetzen möchten.

Wenn Sie Ihr Rohdokument FORMATIEREN, müssen Sie sicherstellen, dass das Feld "Formatierprofil" im "Formatieren *Dokumentname*"-Bildschirm markiert ist. (Der Befehl FORMATIEREN ist im *Con-nect Benutzerhandbuch* im Abschnitt *Dokumente* beschrieben.)

14

Spezielle Zeichen

▪ @	285
▪ Apostroph (')	285
▪ Dollar (\$)	285
▪ Kaufmännisches Und (&)	286
▪ Komma (,)	287
▪ Leerzeichen	287
▪ Nummernzeichen (#)	287
▪ Punkt (.)	289
▪ Schrägstrich (/)	289
▪ Semikolon (;)	289
▪ Zirkumflex (^)	289

Bestimmte Zeichen haben in Con-form eine spezielle Bedeutung. In einigen Fällen können Sie die spezielle Bedeutung einem anderen Zeichen zuordnen. Die anfänglich vorgegebenen Zeichen und deren Bedeutung sind in diesem Abschnitt beschrieben.

Dieses Kapitel behandelt die folgenden Themen:

@

Anfänglich ist das Symbol @ als Umschaltverriegelungszeichen definiert (schaltet von Kleinschreibung auf Großschreibung um). Das Umschaltverriegelungszeichen ist nur wirksam, wenn die Kleinschreibung eingeschaltet ist (.LO ON).

Sie können ein anderes Zeichen für diesen Zweck definieren, z.B. das Nummernzeichen:

```
.OP LOC=#
```

Apostroph (')

Wenn ein Parameter Leerzeichen oder Kommas enthält, müssen Sie ihn in Apostrophe einschließen.

Wenn eine Konstante in den Anweisungen .IF oder .WH Leerzeichen oder Kommas enthält, müssen Sie sie in Apostrophe einschließen.

Dollar (\$)

Anfänglich ist das Dollarzeichen als Zeilenendezeichen definiert. Es wird nur als Zeilenendezeichen interpretiert, wenn es das letzte Zeichen einer Rohertextzeile ist.

Sie können ein anderes Zeichen als Zeilenendezeichen definieren, z.B. das Prozentzeichen. Dies ist mit einer der beiden folgenden Anweisungen möglich:

```
.OP END=%
```

```
.EC %
```

Kaufmännisches Und (&)

Anfänglich wird das kaufmännische Und als Variablenzeichen benutzt.

Sie können ein anderes Zeichen als Variablenzeichen definieren, z.B. das Paragrafenzeichen:

```
.OP VSG=§
```

Um die Wirkung des Variablenzeichens (d.h. die Ersetzung) auszuschalten, geben Sie an:

```
.SU OFF
```

Komma (,)

Das Komma trennt die Parameter einer Anweisung.

Wenn ein Makro mehrere Parameter hat, müssen Sie zwischen den Parametern jeweils ein Komma oder Leerzeichen angeben.

Wenn eine Konstante in den Anweisungen .IF oder .WH Kommas oder Leerzeichen enthält, müssen Sie sie in Apostrophe einschließen.

Leerzeichen

Ein oder mehrere Leerzeichen am Anfang einer Rohtextzeile bewirken einen Zeilenumbruch. Im formatierten Text werden die Leerzeichen am Zeilenanfang mitausgegeben.

Wenn eine Anweisung oder ein Makro einen oder mehrere Parameter hat, müssen Sie zwischen der Anweisung bzw. dem Makro und dem ersten Parameter ein Leerzeichen angeben. Wenn ein Makro mehrere Parameter hat, müssen Sie die Parameter trennen, indem Sie jeweils ein Leerzeichen oder Komma dazwischen angeben.

Wenn ein Parameter Leerzeichen oder Kommas enthält, müssen Sie ihn in Apostrophe einschließen.

Bei den Anweisungen .IF und .WH müssen Sie mindestens ein Leerzeichen vor und nach dem Vergleichsoperator angeben.

Wenn eine Konstante in den Anweisungen .IF oder .WH Leerzeichen oder Kommas enthält, müssen Sie sie in Apostrophe einschließen.

Nummernzeichen (#)

Anfänglich wird das Nummernzeichen als Zeichen für die Seitenzahl benutzt. Sie können es in Titel- und Fußzeilen sowie in der Anweisung .SV benutzen.

Das Nummernzeichen repräsentiert immer die aktuelle Seitenzahl, bis auf folgende Ausnahme: Wenn Sie es in der Fußzeile verwenden, steht es entweder für die aktuelle Seitenzahl (.OP PAG=EQU) oder für die nächste Seitenzahl (.OP PAG=DIF).

Sie können ein anderes Zeichen für die Seitenzahl definieren, z.B. das Ausrufezeichen:

.OP PNS=!

Punkt (.)

Ein Punkt in der ersten Spalte einer Rohertextzeile gibt an, dass es sich um eine Con-form-Anweisung handelt.

Um die Wirkung des Punktes auszuschalten, so dass ein Punkt in der ersten Spalte keine Con-form-Anweisung einleitet, geben Sie an:

```
.IC ON
```

Sie können den Punkt auch benutzen, um nach einer Variablen eine andere Variable oder Text ohne Leerzeichen dazwischen auszugeben (z.B. &var1.&var2).

Schrägstrich (/)

Der Schrägstrich wird als Trennzeichen in Titel- und Fußzeilen benutzt. Er trennt die drei Zeilenbereiche für linksbündigen, zentrierten und rechtsbündigen Text. Die Texte, die Sie für die drei Zeilenbereiche definieren, dürfen keinen Schrägstrich enthalten.

Semikolon (;)

Anfänglich wird das Semikolon als Anweisungstrennzeichen benutzt.

Sie können ein anderes Zeichen als Anweisungstrennzeichen definieren, z.B. den Doppelpunkt:

```
.OP CSE=:
```

Zirkumflex (^)

Anfänglich ist der Zirkumflex als Umschaltzeichen für Groß-/Kleinschreibung definiert. Dieses Zeichen ist nur wirksam, wenn die Kleinschreibung eingeschaltet ist (.LO ON).

Sie können ein anderes Zeichen für diesen Zweck definieren, z.B. den Stern:

.OP SHI=*

15 Serienbriefe

▪ Das Statement COMPOSE	292
▪ Den Serienbrief testen	294

Mit Con-form können Sie Daten aus Adabas-Dateien extrahieren, d.h. automatisch Dokumente erstellen, die die extrahierten Daten enthalten. Dazu müssen Sie ein Natural-Programm schreiben, das das Statement COMPOSE benutzt (siehe Natural-Dokumentation).

Dieses Kapitel beschreibt die folgenden Themen:

Das Statement COMPOSE

Das folgende Beispiel zeigt eine von mehreren Möglichkeiten einen Serienbrief anzulegen. Es wird angenommen, dass Sie mit Con-nect auf einem Mainframe arbeiten (siehe *Con-nect Benutzerhandbuch*, Abschnitte über Prozeduren und Dokumente).

Um einen Serienbrief zu erstellen, können Sie folgendermaßen vorgehen:

1. Schreiben Sie ein Natural-Programm, das Daten aus der Datenbank liest. Dieses Programm muss den Namen der Adabas-Datei sowie die Namen der Felder enthalten, die aus dieser Datei benutzt werden. Es muss außerdem das Statement COMPOSE ASSIGNING enthalten, wie z.B.:

```
COMPOSE ASSIGNING
  'ANREDE' = ANREDE
  'NAME' = NACHNAME
  'STRASSE' = STRASSE
  'STADT' = WOHNORT
```

Die ASSIGNING-Klausel weist den Con-form-Variablen, die in Ihrem Dokument vorkommen, Werte zu. Die Operanden links vom Gleichheitszeichen (in Apostrophe eingeschlossen) sind die Variablennamen, die Sie in Ihrem Dokument verwenden. Die Operanden rechts vom Gleichheitszeichen sind die Felder, die aus der Adabas-Datei extrahiert werden.

2. Legen Sie in Con-nect mit der Befehlsfolge "Dokument EINGEBEN" ein Dokument an. Dies ist das Skelett für Ihren Serienbrief. Es enthält den Text und die Variablen, die mit den Daten aus der Adabas-Datei ersetzt werden sollen.

```
.LM 0;.RM 60
&anrede &name
&strasse
&stadt
.SL
Sehr geehrte/r &anrede &name.,
.SL
.LM 0;.RM 60
Ihr Abonnement unseres Magazins läuft bald aus.
Wenn Sie nun handeln und das Abonnement um ein weiteres Jahr
verlängern, bieten wir Ihnen einen Rabatt von 40% - Sie sparen
```



```
DM 38,00 !  
.GRUSS
```



Anmerkung: Das Makro .GRUSS im obigen Beispiel wurde im Formatierprofil FPROFILE definiert.

3. Benutzen Sie die Befehlsfolge "Prozedur EINGEBEN", um das Natural-Programm in Con-nect zu definieren.
4. Benutzen Sie die Befehlsfolge "Dokument *dokumentname* INFO", um die Prozedur mit dem Con-nect-Dokument zu verknüpfen. Definieren Sie die Prozedur als Versorgungsprozedur.
5. FORMATIEREN Sie das Dokument.



Vorsicht: Wenn Sie den Befehl FORMATIEREN in Con-nect benutzen, darf die Prozedur keine COMPOSE-Statements mit anderen Klauseln als ASSIGNING oder EXTRACTING enthalten.

Den Serienbrief testen

Das folgende Beispiel für einen Serienbrief liefert verschiedene Ergebnisse, je nachdem, ob der Empfänger weiblich oder männlich ist. Das Beispiel zeigt auch den Vorteil der Anweisung `.SV`. Diese Anweisung ist hilfreich, wenn Sie testen möchten, ob der Serienbrief wie gewünscht aussieht, bevor Sie die Variablen mit Daten aus der Datenbank ersetzen lassen.

```
.LM 0;.RM 60
.FI ON;.JU ON
.SV geschlecht=W
.SV geehrt=geehrte
.SV anrede=Frau
.SV name=Schmitt
.SV strasse=Bergstr. 99
.SV stadt=22220 Bergheim
.SV magname=NATURWUNDER
.** Die oben definierten Variablen sind hilfreich, wenn Sie testen
.** möchten, ob der Serienbrief richtig ausgegeben wird. Um die
.** Variablen mit Daten aus der Datenbank zu ersetzen, müssen Sie
.** ein Natural-Programm schreiben, das das Statement COMPOSE benutzt.
.** Wenn Sie die Variablen mit Daten aus der Datenbank ersetzen
.** wollen, müssen Sie die obigen .SV-Anweisungen löschen.
.IF &geschlecht = M
.TH
.SV magneu=SPORT-MAGAZIN
.SV text1=Höhepunkte der Olympiade
.SV text2=Neues über die Medaillengewinner
.SV durchwahl=334450
.EL
.SV magneu=UMWELT & GESUNDHEIT
.SV text1=Neueste Erkenntnisse aus der Ernährungsforschung
.SV text2=Rezepte für Vegetarier
.SV durchwahl=334451
.EI
&anrede &name.$
&strasse.$
&stadt.$
.SL 3
.RA ON
&$DA.. &$MG &$CN.&$YE
.RA OFF
.SL 3
Sehr &geehrt &anrede &name.,
.SL
Ihr Abonnement des Magazins &magname läuft demnächst aus. Wenn Sie
nun handeln und das Abonnement um ein weiteres Jahr verlängern,
erhalten Sie einen Rabatt von 40% - Sie sparen 38,00 DM!
Lassen Sie sich dieses Angebot nicht entgehen!
```

.SL

Außerdem möchten wir Ihnen unser neuestes Magazin &magneu vorstellen, das folgende Themen behandelt: &text1 und &text2..

Wenn Sie an &magneu interessiert sind, füllen Sie einfach den beigefügten Bestellschein aus oder rufen Sie gebührenfrei an unter 101-202-&durchwahl..

.SL

Mit freundlichen Grüßen

.SL 4

J. Becker\$

Vertriebsleiter

Die formatierte Ausgabe sieht folgendermaßen aus:

Frau Schmitt
Bergstr. 99
22220 Bergheim

19. Juli 2000

Sehr geehrte Frau Schmitt,

Ihr Abonnement des Magazins NATURWUNDER läuft demnächst aus. Wenn Sie nun handeln und das Abonnement um ein weiteres Jahr verlängern, erhalten Sie einen Rabatt von 40% - Sie sparen 38,00 DM! Lassen Sie sich dieses Angebot nicht entgehen!

Außerdem möchten wir Ihnen unser neuestes Magazin UMWELT & GESUNDHEIT vorstellen, das folgende Themen behandelt: Neueste Erkenntnisse aus der Ernährungsforschung und Rezepte für Vegetarier. Wenn Sie an UMWELT & GESUNDHEIT interessiert sind, füllen Sie einfach den beigefügten Bestellschein aus oder rufen Sie gebührenfrei an unter 101-202-334451.

Mit freundlichen Grüßen

J. Becker
Vertriebsleiter

Wenn die Variablen "geschlecht", "geehrt" und "anrede" aus dem obigen Con-form-Dokument die folgenden Werte enthalten, sieht die formatierte Ausgabe anders aus:

```
.SV geschlecht=M  
.SV geehrt=geehrter  
.SV anrede=Herr
```

In diesem Fall sieht die formatierte Ausgabe folgendermaßen aus:

Herr Schmitt
Bergstr. 99
22220 Bergheim

19. Juli 2000

Sehr geehrter Herr Schmitt,

Ihr Abonnement des Magazins NATURWUNDER läuft demnächst aus. Wenn Sie nun handeln und das Abonnement um ein weiteres Jahr verlängern, erhalten Sie einen Rabatt von 40% - Sie sparen 38,00 DM! Lassen Sie sich dieses Angebot nicht entgehen!

Außerdem möchten wir Ihnen unser neuestes Magazin SPORT-MAGAZIN vorstellen, das folgende Themen behandelt: Höhepunkte der Olympiade und Neues über die Medaillengewinner. Wenn Sie an SPORT-MAGAZIN interessiert sind, füllen Sie einfach den beigefügten Bestellschein aus oder rufen Sie gebührenfrei an unter 101-202-334450.

Mit freundlichen Grüßen

J. Becker
Vertriebsleiter

16

Veraltete Anweisungen

▪ .>> - Sprungsymbol	299
▪ .GO - Sprung	299
▪ .LO - Kleinschreibung	300
▪ .OP LOC - Umschaltverriegelungszeichen	300
▪ .OP SHI - Umschaltzeichen für Groß-/Kleinschreibung	302
▪ .PN - Seitenzahl	302
▪ .TY - Text auf Bildschirm ausgeben	302
▪ .UP - Großschreibung	303

Dieses Kapitel listet alle Con-form-Anweisungen in alphabetischer Reihenfolge auf, die veraltet sind und nur noch aus Gründen der Kompatibilität zu früheren Con-form-Versionen verfügbar sind. Es wird empfohlen, diese Anweisungen *nicht* zu benutzen.

.>> - Sprungsymbol

```
.>> sprungsymbol
```

Diese Anweisung definiert das Sprungziel für die Anweisung `.GO`. Das Sprungsymbol muss innerhalb einer Schleife oder eines Makros eindeutig sein.

Um z.B. "ende" als Sprungsymbol zu definieren, geben Sie an:

```
.>> ende
```

.GO - Sprung

```
.GO sprungsymbol
```

Die Anweisungen `.GO` und `.>>` bilden ein Paar. Sie können nur in einem Makro oder in einer While-Schleife benutzt werden.

Die Anweisung `.GO` wird benutzt, um die Verarbeitung an dem Sprungziel fortzusetzen, das mit der Anweisung `.>>` definiert wurde. Das Sprungziel kann vor oder nach der Anweisung `.GO` stehen.

Um z.B. zum Sprungsymbol "ende" zu springen, geben Sie an:

```
.GO ende
```

.LO - Kleinschreibung

```
.LO ON  
.LO OFF
```

Anfänglich ist die Kleinschreibung ausgeschaltet, d.h. alle Zeichen, die als Großbuchstaben eingegeben wurden, werden auch als Großbuchstaben ausgegeben. Dies entspricht der Anweisung:

```
.LO OFF
```

Wenn Ihr Terminal Kleinbuchstaben nicht unterstützt, so dass Sie nur Großbuchstaben eingeben können, können Sie die Textabschnitte definieren, die Sie in Kleinbuchstaben ausgegeben möchten. Benutzen Sie dazu die Anweisung:

```
.LO ON
```

Daraufhin werden alle Buchstaben in Kleinbuchstaben umgewandelt.

Die folgenden Ausnahmen gelten für die Anweisung `.LO`, d.h. folgende Buchstaben werden nicht in Kleinbuchstaben umgewandelt:

- der Buchstabe nach dem Umschaltzeichen für Groß-/Kleinschreibung
- alle Buchstaben, die zwischen zwei Umschaltverriegelungszeichen stehen (siehe unten).

.OP LOC - Umschaltverriegelungszeichen

```
.OP LOC=zeichen
```

Diese Anweisung wird zusammen mit der Anweisung `.LO ON` benutzt, wenn der Rohertext in Großbuchstaben vorliegt. Sie können Buchstaben zwischen zwei Umschaltverriegelungszeichen einschließen, damit sie als Großbuchstaben ausgegeben werden, d.h. damit sie nicht in Kleinbuchstaben umgewandelt werden.

Das Umschaltverriegelungszeichen muss in einer Rohertextzeile jeweils paarweise vorkommen. Um größere Textabschnitte in Großbuchstaben auszugeben, ist es sinnvoller, die Anweisung `.LO OFF` zu verwenden.

Anfänglich ist das Zeichen `@` das Umschaltverriegelungszeichen. Um ein anderes Zeichen dafür zu definieren, z.B. das Nummernzeichen (`#`), geben Sie an:

.OP LOC=~~#~~

.OP SHI - Umschaltzeichen für Groß-/Kleinschreibung

```
.OP SHI=zeichen
```

Diese Anweisung wird zusammen mit der Anweisung `.LO ON` benutzt, wenn der Rohtext in Großbuchstaben vorliegt. Sie können dieses Zeichen vor einem Buchstaben eingeben, damit er als Großbuchstabe ausgegeben wird, d.h. damit er nicht in einen Kleinbuchstaben umgewandelt wird.

Anfänglich ist der Zirkumflex (^) das Umschaltzeichen für Groß-/Kleinschreibung. Um ein anderes Zeichen dafür zu definieren, z.B. den Stern (*), geben Sie an:

```
.OP SHI=*
```

.PN - Seitenzahl

```
.PN zahl
```

Diese Anweisung definiert eine Seitenzahl für die nächste Seite. Sie bewirkt keinen Seitenumbruch. Es wird empfohlen, stattdessen die Anweisung `.NP` zu verwenden.

.TY - Text auf Bildschirm ausgeben

```
.TY text
```

Sie können in dem Bereich, der für Con-nect-Meldungen reserviert ist, Informationen ausgeben. Geben Sie dazu den gewünschten Text als Parameter der Anweisung `.TY` an, z.B.:

```
.TY Dies ist Seite 5 des Rohtextes.
```

Wenn Sie in dem Text das Anweisungstrennzeichen (anfänglich das Semikolon) benutzen möchten, müssen Sie es doppelt angeben (;:). Wenn Sie die Anweisung `.TY` ohne Parameter angeben, erscheint eine Leerzeile.

Bevor der als Parameter angegebene Text ausgegeben wird, wird Folgendes für diesen Text durchgeführt: Variablenersetzung (siehe die Anweisung `.SV`), Zeichenumsetzungen (siehe die Anweisungen `.TR` und `.TS`) sowie Umwandlung von Groß-/Kleinschreibung (siehe die Anweisungen `.UP` und `.LO`).

.UP - Großschreibung

```
.UP ON  
.UP OFF
```

Wenn Sie Text in Kleinbuchstaben eingegeben haben, können Sie ihn in Großbuchstaben umwandeln.

Um einen Textabschnitt in Großbuchstaben auszugeben, müssen Sie ihn zwischen den beiden folgenden Anweisungen angeben:

```
.UP ON
```

```
.UP OFF
```

Die Anweisung `.UP` bewirkt keinen Zeilenumbruch.

