

## Übersetzung der Dokumentation

# Das csquotes-Paket (Version 5.1) Kontextsensitive Zitatanlage

übersetzt von Theresia Conrad  
Patrik Faßbender and Christine Römer  
(patristo19@gmail.com, christine-roemer@t-online.de)

Mai 2011 / Version 1.01 / Lizenz lppl

## Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>1</b>	<b>7 Hilfskommentare</b>	<b>17</b>
<b>1 Einleitung</b>	<b>2</b>	<b>8 Konfiguration</b>	<b>18</b>
<b>2 Paketoptionen</b>	<b>2</b>	<b>9 Hinweise zur Anwendung</b>	<b>28</b>
<b>3 Basisschnittstellen</b>	<b>6</b>	<b>10 Hinweise und Vorsorge</b>	<b>33</b>
<b>4 Aktivzitate</b>	<b>10</b>	<b>11 Autorenschnittstelle</b>	<b>38</b>
<b>5 Integrierte Schnittstellen</b>	<b>13</b>	<b>12 Geschichte der Revision</b>	<b>43</b>
<b>6 Darstellungsumgebungen</b>	<b>15</b>		

## Tabellenverzeichnis

1 Sprachoptionen . . . . .	4	4 Konfigurierbare Parameter . . .	21
2 Designs und Varianten . . . . .	18	5 Zusatzhaken . . . . .	23
3 Sprachsynonyme . . . . .	20	6 Anführungszeichen . . . . .	34

# 1 Einleitung

## 1.1 Über `csquotes`

Dieses Paket stellt fortgeschrittene Funktionen bereit, mit denen Zitate angezeigt und im Inline-Modus verwendet werden können. Es ist für einen großen Bereich an Aufgaben erstellt worden, von den einfachsten Anwendungen bis hin zu den komplexen Anforderungen formeller Zitierungen. Die Anlagen beinhalten Anweisungen, Umgebungen und Anwender-definierbare ‚intelligente Zitierungen‘ welche sich dynamisch an ihren Inhalt anpassen. Die Anführungszeichen wechseln automatisch im Fall, dass die Zitate verschachtelt sind, und passen sich ebenso automatisch der aktuellen Sprache an. Überdies enthält das Paket zusätzliche Möglichkeiten, die erstellt wurden, um mit den besonderen Anforderungen von akademischen Schriftstücken zurechtzukommen. Alle Zitatdesigns und selbstverständlich auch die optionalen aktiven Zitate sind frei selbst-konfigurierbar.

## 1.2 Lizenz

Copyright © 2003–2010 Philipp Lehman. Permission is granted to copy, distribute and/or modify this software under the terms of the LaTeX Project Public License, version 1.3c.<sup>1</sup> This package is author-maintained.

## 1.3 Beiträge

Die multilinguale Unterstützung hängt von den Beiträgen der Anwender ab. Wenn auch Sie etwas beisteuern wollen, zu einem Zitatdesign, dann möchte ich Sie bitten, dass Sie sich § 8.1 anschauen, um einen Überblick über alle Komponenten ihres Zitatdesign und deren Zusammenhängen zu bekommen. Tabelle (Tabelle 6) ist für eine Auflistung von Kommandos, welche genutzt werden können, um Zitatdesigns zu definieren.

# 2 Paketoptionen

Alle Paketoptionen werden in der  $\langle key \rangle = \langle value \rangle$  [‚Schlüssel‘=‚Wert‘] Syntax angegeben.

---

<sup>1</sup> <http://www.ctan.org/tex-archive/macros/latex/base/lppl.txt>

## 2.1 Die Hauptoptionen

`strict=true, false`

default: false

Diese Option wandelt alle Paketwarnungen in Fehlermeldungen um. Das ist immer dann hilfreich, wenn Sie gewährleisten wollen, dass kein Fehler bei der Fertigstellung eines Dokuments unerkannt bleibt. Die Kurzform `strict` ist äquivalent zu `strict=true`.

`style=<style>`

default: american

Diese Option wählt ein fertiges Zitatdesign aus. Dieses Design wird nun fortan im Dokument verwendet, bis dieses manuell geändert wird, siehe § 3.7 für genauere Informationen. Diese Option impliziert Paare von `autostyle=false`. Bitte beziehen Sie sich dabei auf Tabellen 2 und 3 als eine Liste von verfügbaren Zitatdesigns und deren Synonyme. Siehe §§ 8.1, 8.2, 9.1 für Anweisungen bezüglich des Hinzufügens und Modifizierens von Zitatdesigns.

`autostyle=true, false, try, once, tryonce`

default: tryonce

Diese Option steuert die multilinguale Unterstützung. Es benötigt dafür das `babel`-Paket oder das `polyglossia`-Paket.<sup>2</sup> `autostyle=true` Diese Anweisung passt das Zitatdesign so an, dass es zur momentan im Dokument verwendeten Sprache passt; `once` ändert das Zitatdesign einmalig, sodass es zur Grundsprache des Dokumentes passt. `autostyle=try` und `tryonce` verhalten sich, sollte multilinguale Unterstützung vorhanden sein, wie `true` und `once`, doch werden sie, für den Fall, dass dies nicht möglich wäre, keine Warnungen erzeugen. (D.h. falls weder `babel` noch `polyglossia` geladen wurden). Die Kurzform `autostyle` ist äquivalent zu `autostyle=true`. Siehe auch § 3.7.

`<language>=<variant>`

Benutzen Sie die Sprachoptionen aus der Tabelle (Tabelle 1), um eine Designvariante auszuwählen, falls dort mehr als eine für eine bestimmte Sprache vorhanden sein sollte. Die erste Variante in der Liste ist jeweils die Voreinstellung für das jeweilige Design. Im folgenden Beispiel macht die `autostyle`-Option, dass `csquotes` ständig das Zitatdesign der momentan verwendeten Sprache anpasst, wobei stets die voreingestellten Varianten für die Sprachen verwendet werden. Die Voreinstellungen von Deutsch und Norwegisch wurden geändert:

```
\usepackage[english,ngerman]{babel}
\usepackage[autostyle,german=guillemets,norwegian=quotes]{csquotes}
```

<sup>2</sup> Notiere, `polyglossia`-Unterstützung ist zur Zeit noch in einem Vorbereitungsstadium, denn `polyglossia` fehlt es an einer Schnittstelle zu anderen Paketen. In der Anwendung führt das dazu, dass `csquotes` fähig ist, (e.g., `english`) zu erkennen, jedoch nicht die konkrete Sprachvariante (e.g., `british`).

Optionsschlüssel	mögliche Werte
croatian (kroatisch)	quotes, guillemets, guillemets*
danish (dänisch)	quotes, guillemets
english (englisch)	american, british
french (französisch)	quotes, quotes*, guillemets, guillemets*
german (deutsch)	quotes, guillemets, swiss
italian (italienisch)	quotes, guillemets
norwegian (norwegisch)	guillemets, quotes
portuguese (portugiesisch)	portuguese, brazilian
spanish (spanisch)	spanish, mexican
swedish (schwedisch)	quotes, guillemets, guillemets*

Tabelle 1: Sprachoptionen definiert durch Voreinstellungen

Man bemerke, dass der babel-Sprachname `ngerman` ist, wohingegen `csquotes german` als Sprachnamen verwendet. Immer wenn ein Zitatdesign automatisch ausgewählt wird, wird dieses Paket zuerst die Sprachnamen normalisieren, unter Verwendung einer Liste von Synonymen, welche die Sprache dem Zitatdesign zuordnen. Siehe Tabelle 3 für detailliertere Informationen. Tabelle 1 zeigt die Sprachoptionen, die als voreingestellt definiert sind. Zusätzliche Optionen müssten in der Konfigurationsdatei definiert werden. Siehe §§ 8.3 and 9.1. Auch hilfreich ist § 10.9, denn es enthält einige zusätzliche Informationen bezüglich der vordefinierten Designs.

`maxlevel`= $\langle integer \rangle$  default: 2

Diese Option kontrolliert den maximalen Verschachtlungsgrad von Zitaten. In der Voreinstellung unterstützt diese Paket zwei Grade von Zitaten, äußere und innere, und verursacht einen Fehler, sollten Zitate darüber hinaus verschachtelt sein. Alternativ können auch innere und äußere Zitate wieder verwertet werden für weitere Zitatlevel. Diese alternative Variante ist nicht möglich, sollte der maximal mögliche Zitatgrad `maxlevel` erhöht worden sein. Das Minimum ist 2.

`autopunct`=`true`, `false` default: `true`

Diese Option reguliert, ob die Zitatbefehle vorausschauend auf Satzzeichen reagieren. Siehe §§ 3.3, 3.5, 5.1, 5.3 für Details. Siehe auch `\DeclareAutoPunct` § 8.9 und § 8.7, um zu erfahren, wie die vorausschauende Funktion reguliert und verwendet wird. Die Kurzform `autopunct` ist identisch mit `autopunct=true`.

`threshold`= $\langle integer \rangle$  default: 3

Diese Option definiert einen Schwellenwert für die Anzahl der Zeilen oder Wörter von Blockzitate-Anlagen, um zu bestimmen, ob ein Zitat im Inline- oder im Anzeigemodus geschrieben werden muss. Es entspricht dem `\SetBlockThreshold`-Befehl. Siehe §§ 3.5 and 8.6 für weitere Details.

`thresholdtype=lines, words`

default: lines

Diese Option wählt den Block-Schwellenwert-Typ. Mit `thresholdtype=lines` wird festgelegt, wie viele Zeilen nötig sind, um das Zitat zu setzen; mit `thresholdtype=words` wird die Anzahl der Wörter im Zitat als Schwellenwert gezählt.<sup>3</sup> Die Voreinstellung des Schwellenwertes beträgt 3 Zeilen. Sollten Sie beispielsweise 50 Wörter als Schwellenwert bevorzugen, setzen Sie `threshold=50` und `thresholdtype=words`. Siehe § 3.5 für weitere Details.

`parthreshold=true, false`

default: true

Diese Option erlaubt eine feinere Einstellung des Block-Schwellenwertes. Wenn sie aktiviert ist, wird jeder explizite Absatz oder Zeilenumbruch im Zitat den Schwellenwert beeinflussen, d. h., dass das Zitat im Display-Modus mit seinen Zeilenumbrüchen und Absätzen gesetzt wird, unabhängig von seiner tatsächlichen Länge. Bei deaktivierter Option und `thresholdtype=lines` werden explizite Absätze und Zeilenumbrüche normal angewendet. Bei `thresholdtype=words` werden sie als normale Wortgrenze behandelt. Die Kurzform `parthreshold` ist äquivalent zu `parthreshold=true`.

`splitcomp=true, false`

default: true

Diese Option ist nur zusammen mit `thresholdtype=words` anwendbar. Sie ermöglicht es, auch Wortzusammensetzungen mit Bindestrich richtig auszuwerten. Bei Aktivierung werden Zusammensetzungen an allen Bindestrichen geteilt und als separate Wörter gezählt. Bei ausgeschalteter Option werden Zusammensetzungen als einzelnes Wort gezählt. Die Kurzform `splitcomp` ist äquivalent zu `splitcomp=true`.

`csdisplay=true, false`

default: false

In ihrer Voreinstellung werden die Blockzitat-Anlagen Zitate im Inline-Modus erzwingen, wenn sie in Fußnoten, Parboxen, Minipages und Fließkommazahlen angewendet werden. In diesen Fällen erlaubt das Setzen dieser Option auf `true` einen kontextsensitiven Wechsel zu Zitaten im Display-Modus wie im Textkörper. Die Kurzform `csdisplay` ist äquivalent zu `csdisplay=true`. Diese Option kann auch lokal begrenzt eingesetzt werden mit `\csdisplaytrue`, beziehungsweise `\csdisplayfalse`.

`debug=true, false`

default: false

Diese Option reguliert das debugging-Feature von Blockzitat-Anlagen. Aktiviert werden alle Blockzitat-Befehle als diagnostische Nachrichten und in der Protokoll-Datei (transcript file) ausgegeben. Diese Nachrichten zeigen die errechnete (n) Zeilen-/Wörterzahl. Explizite Absätze und Zeilenumbrüche werden in dem Zitattext erkannt

---

<sup>3</sup> Das Wort Zählcode basiert auf einer Idee von Donald Arseneau.

und die finale Auswahl des Modus (Display/Inline) wird festgestellt. Die Kurzform `debug` ist äquivalent zu `debug=true`.

## 2.2 Kompatibilitätsoptionen

`version=<version>`, 4.4, 3.6, 3.0

Diese Option ist ein Versuch, zumindest in gewissem Umfang, `csquotes` abwärtskompatibel mit anderen älteren Versionen zu machen. Momentan wird eine Abwärtskompatibilität unterstützt mit Version 4.4, welche 3.7–4.4 nachahmt, Version 3.6, die 3.1–3.6 nachahmt und Version 3.0. Ältere Versionen werden nicht unterstützt. Es ist möglich, dass auch nicht aufgelistete `<version>` kompatibel sind. In diesem Fall wird das Paket die `<version>`s Nummer hinzufügen, solange es entweder einen passenden Kompatibilitätsmodus oder die aktuelle Versionsnummer findet. Als Beispiel, wenn `version=4.0` angegeben wird, wird `csquotes` die Versionsnummer solange erhöhen bis 4.4 erreicht wird und dann den `v4.4` Kompatibilitätsmodus aktivieren, denn das ist die letzte noch unterstützte Version, die kompatibel ist, mit den Anforderungen von 4.0. Das beinhaltet das `csquotes` möglicherweise aufgerufen wird, in gewissen Maße ‚Zukunftssicher‘, in einer Art und Weise, in der die Version beim Schreiben langsamer arbeiten könnte. Falls zukünftige Versionen die Abwärtskompatibilität nicht unterstützen, werden sie automatisch den besten Kompatibilitätsmodus aktivieren. Andernfalls hat die Option `version` einfach keine Auswirkungen.

`babel=true, false, try, once, tryonce`

Ein älterer Name für die `autostyle`-Option. Diese Option ist veraltet.

## 3 Basisschnittstellen

Das Paket unterstützt zwei Wege, um Zitate zu kennzeichnen: `built-in`-Befehle und Aktivzitate werden in der Präambel des Dokuments oder in der Konfigurationsdatei (configuration file) definiert. Dieses Kapitel wird Sie in die Grundkommandos einweisen, Aktivzitate werden erläutert in § 4. Wenn Sie mit automatischen Anführungszeichen arbeiten, werden Ihnen die integrierten Zitanlagen in § 5 präsentiert.

### 3.1 Regulären Text zitieren

Die grundlegendsten Kommandos umschließen das Argument in Anführungszeichen:

```
\enquote{<text>}
\enquote*{<text>}
```

Wie alle Zitanlagen ist auch dieser Befehl kontextsensitiv. Da es abhängig vom Verschachtlungsgrad ist, wird es bei einfachen und verschachtelten Zitaten zwischen äußeren und inneren Anführungszeichen wechseln. Die Version mit Stern springt direkt zur inneren Ebene. Wenn die multilinguale Unterstützung aktiviert ist, wird das Aussehen aller Anführungszeichen an die aktuelle Sprache angepasst.

### 3.2 Fremdsprachige Zitate

Um Zitate in einer Fremdsprache zu setzen, stellt `csquotes` zwei Befehle bereit. Diese kombinieren `\enquote` mit dem Sprachwechsel vom `babel`- oder `polyglossia`-Paket:

```
\foreignquote{<lang>}{<text>}
\foreignquote*{<lang>}{<text>}
```

Dieser Befehl kombiniert `\enquote` mit `\foreignlanguage`. Es wechselt Silbentrennungsmuster und aktiviert extra Definitionen, die von `babel`/`polyglossia` für `<lang>` bereitgestellt werden. Das muss ein Sprachname sein, der im jeweiligen Paket bekannt sein muss. Die Anführungszeichen werden der Sprache der zitierten Textstelle angepasst.

```
\hyphenquote{<lang>}{<text>}
\hyphenquote*{<lang>}{<text>}
```

Dieser Befehl kombiniert `\enquote` mit der `hyphenrules` Umgebung, das heißt, er wählt lediglich Silbentrennungsmuster. Die Anführungszeichen werden dabei so ausgewählt, dass sie mit der Textsprache um das Zitat übereinstimmen.

### 3.3 Formelle Zitate von regulärem Text

Formalzitate werden typischerweise von einem Hinweis begleitet, der anzeigt, an welcher Stelle die Zitatquelle zu finden ist. Der folgende Befehl ist eine Erweiterung von `\enquote`, die den zitierten Text mit Anführungszeichen umschließt und einen Hinweis hinter dem Zitat ergänzt.

```
\textquote[<cite>][<punct>]{<text>}<tpunct>
\textquote*[<cite>][<punct>]{<text>}<tpunct>
```

Mit `<text>` kann jede beliebige Textstelle in Anführungszeichen eingeschlossen werden. Mit den optionalen Parametern `<cite>` und `<punct>` kann der Hinweis oder die Schlusspunkte vom `<text>` variiert werden. `<tpunct>` kennzeichnet die nachlaufende

Zeichensetzung hinter dem Befehl. Wenn die `autopunct`-Option aktiviert ist, wird der Zitatbefehl die vorrausgehende Zeichensetzung erkennen, die dem letzten Argument folgt, und kann sie, falls erforderlich, bewegen. Siehe § 8.7, um zu erfahren, wie man die Weise ändert, in der diese Argumente gehandhabt werden und § 9.2, um Gründe zu erfahren, warum man die Zeichensetzung als zweites Argument ändern könnte. Die mit einem Stern versehene Variante dieses Befehls springt wiederum direkt zur inneren Zitatebene. Hier einige Anwendungsbeispiele:

```
\textquote{...}
\textquote[.]{...}
\textquote[Doe 1990, 67]{...}
\textquote[{\cite[67]{doe90}}]{...}
```

Beachten Sie die Verwendung der optionalen Argumente in den obigen Beispielen. Wie Sie im zweiten Beispiel sehen, ist `<cite>` immer dann erforderlich, wenn auch `<punct>` verwendet wird, auch dann, wenn diese Option leer ist. Beachten Sie außerdem, dass ein optionales Argument mit eckigen Klammern in einem zusätzlichen Paar von geschweiften Klammern verpackt werden muss, wie im letzten Beispiel angezeigt. Beim Arbeiten mit automatisierten Zitaten möchten Sie vielleicht auch über die integrierten Zitanlagen, dargestellt in § 5, etwas lernen.

### 3.4 Formalzitate in einer Fremdsprache

Es existieren zwei zusätzliche Befehle, welche eine Verbindung zwischen `\textquote` und der Sprachauswahl von `babel` oder dem `polyglossia`-Paket bereitstellen:

```
\foreigntextquote{<lang>}[<cite>][<punct>]{<text>}<tpunct>
\foreigntextquote*{<lang>}[<cite>][<punct>]{<text>}<tpunct>
```

Dieser Befehl verbindet `\textquote` mit `\foreignlanguage`. Getrennt von der Sprache werden die Argumente durch `\textquote` gehandhabt.

```
\hyphentextquote{<lang>}[<cite>][<punct>]{<text>}<tpunct>
\hyphentextquote*{<lang>}[<cite>][<punct>]{<text>}<tpunct>
```

Dieser Befehl verbindet `\textquote` mit der `hyphenrules` Umgebung. Im Gegensatz zur Sprache werden die Argumente gehandhabt wie in `\textquote`.

### 3.5 Zitatblöcke von normalem Text

Formale Anforderungen an wissenschaftliches Schreiben verlangen häufig, dass Zitate in den Text eingebettet werden, solange diese kurz sind. Aber wenn sie eine bestimmte Anzahl von Zeilen oder Wörtern überschreiten, dann müssen diese in der Regel einen eingerückten Absatz bilden, ein sogenanntes Blockzitat. Im zweiten

Fall müssen auch keine Anführungszeichen gesetzt werden. Der nun folgende Befehl befasst sich mit diesen Anforderungen und regelt die Auswahl automatisch:

```
\blockquote[⟨cite⟩][⟨punct⟩]{⟨text⟩}⟨tpunct⟩
```

Dieser Befehl legt die Länge des `⟨text⟩` fest. Sollte die Länge einen festgelegten Schwellenwert überschreiten, wird das Zitat im Display-Modus gesetzt, d. h. als Zitatblock. Wenn sie den Wert nicht überschreitet, wird sich das `\blockquote` wie `\textquote` verhalten. In Anlehnung an die `thresholdtype`-Option kann auch hier der Schwellenwert von einer Anzahl von Zeilen oder von Wörtern im `⟨text⟩` abhängen. Wenn die `parthreshold`-Option aktiviert ist, wird jeder wesentlich gesetzte Absatz und Zeilenumbruch im `⟨text⟩` Einfluss nehmen auf den Schwellenwert, d. h. es wird unabhängig seiner Länge im Display-Modus gesetzt werden. Die Voreinstellung des Schwellenwertes bei Zeilen liegt bei 3 Zeilen, wenn `parthreshold` aktiviert ist. Die voreingestellte Umgebung, die für Display-Zitate verwendet wird, ist `quote`. Siehe §§ 2.1 und 8.6 um diese Parameter zu verändern. Beachte, dass `csquotes` in der Voreinstellung Zitate im Inline-Modus in Fußnoten, Parboxen, Minipages und Fließkommazahlen erzwingen wird. Verwenden Sie die `csdisplay`-Option aus § 2.1, um dieses Verhalten zu ändern. Die optionalen Argumente `⟨cite⟩` und `⟨punct⟩` geben das Zitat und die endgültige Zeichensetzung des `⟨text⟩` an. `⟨tpunct⟩` markiert nachfolgende Zeichensetzung(. . .) nach dem Befehl. Die `autopunct`-Option reguliert, ob die Zitierbefehle vorausschauend auf Satzzeichen reagieren und diese, wenn nötig, bewegen. Siehe § 8.7, um zu erfahren, wie man die Weise ändert, in der diese Argumente gehandhabt werden und § 9.2, um Gründe zu erfahren, warum man die Zeichensetzung als zweites Argument ändern könnte.

### 3.6 Blockzitate von Texten in einer Fremdsprache

Die folgende Befehle verbinden `\blockquote` mit den Sprachwechselfähigkeiten der Pakete von `babel` oder `polyglossia`:

```
\foreignblockquote{⟨lang⟩}[⟨cite⟩][⟨punct⟩]{⟨text⟩}⟨tpunct⟩
```

Dieser Befehl verhält sich wie ein `\foreignquote`, solange das Zitat kurz genug ist. Wenn das Zitat den Schwellenwert überschreitet, wird es eingehüllt in eine `otherlanguage*`-Umgebung, welche wiederum in eine Blockzitatumgebung gesetzt wird. Die Argumente werden analog zu `\blockquote` gehandhabt.

```
\hyphenblockquote{⟨lang⟩}[⟨cite⟩][⟨punct⟩]{⟨text⟩}⟨tpunct⟩
```

Dieser Befehl verhält sich wie `\hyphenquote`, wenn das Zitat kurz genug ist. Wenn das Zitat den Schwellenwert überschreitet, wird es eingehüllt in eine `hyphenrules` Umgebung, welche wiederum in eine Blockzitatumgebung gesetzt wird. Die Argumente werden analog zu `\blockquote` gehandhabt.

```
\hybridblockquote{<lang>}[<cite>][<punct>]{<text>}<tpunct>
```

Dieser Befehl verhält sich wie `\hyphenquote`, wenn das Zitat kurz genug ist. Wenn das Zitat den Schwellenwert überschreitet, wird es behandelt wie `\foreignblockquote`. Mit anderen Worten, der Befehl verbindet die Features von `\foreignblockquote` und `\hyphenblockquote`. Die Argumente werden analog zu `\blockquote` gehandhabt.

### 3.7 Zitatdesigns auswählen

Zitatdesigns können überall mit dem nun folgenden Befehl im Dokument (Dokumentkörper) manuell ausgewählt werden:

```
\setquotestyle[<variant>]{<style>}  
\setquotestyle{<alias>}  
\setquotestyle*
```

Die Normalform des Befehls wählt ein Zitatdesign und deaktiviert die vielsprachige Unterstützung. Das notwendige Argument kann ein Zitatdesign oder dessen Synonym sein. Wenn es ein Zitatdesign ist, legt das optionale Argument die Designvariante fest. Der Befehl mit Stern ohne Argumente aktiviert die multilinguale Unterstützung. Siehe Tabellen 2 and 3, eine Liste der möglichen Designs, Designvarianten und Sprachsynonymen.

## 4 Aktivzitate

Das Paket unterstützt auch die Aktivtypen der Befehle, die zu den in § 3 vorgestellten korrespondieren. Grob gesagt, sind diese Aktivtypen Einzeltypen, die wie Befehle funktionieren. Wie auch die korrespondierenden Befehlssequenzen sind die Aktivzitate vollwertige Anzeigeelemente, die anzeigen, wie verschachtelt Zitate sind, und Fehler ausgeben, sollten sie ungültig verschachtelt sein. Wenn die multilinguale Unterstützung aktiviert ist, wird das Aussehen aller Anführungszeichen an die aktuell verwendete Sprache angepasst. Die nun folgenden Befehle verwenden solche Aktivzitate. Sie können in der Konfigurationsdatei, der Präambel oder dem Dokumentenkörper verwendet werden. Man bemerke, dass alle Typen automatisch auf Richtigkeit untersucht werden, sobald sie verwendet werden. Dieses Paket weist alle Typen ab, die nicht als Aktivzitate verwendet werden können. Siehe § 10.3, um Typen zu erfahren, die als Aktivzitate verwendet werden können.

## 4.1 Regulären Text zitieren

`\MakeOuterQuote` und `\MakeInnerQuote` definieren Aktivzitate, welche die inneren und äußeren Anführungszeichen setzen. Beide benötigen ein notwendiges Argument, das Zeichen für die öffnende und abschließende Markierung des Zitats:

```
\MakeOuterQuote{⟨character⟩}
```

```
\MakeInnerQuote{⟨character⟩}
```

`\MakeAutoQuote` definiert Aktivzitate, die automatisch zwischen inneren und äußeren Zitaten wechseln. Die zwei notwendigen Argumente dienen als Öffnungs- und Abschlussmarkierung und müssen eindeutig sein:

```
\MakeAutoQuote{⟨character 1⟩}{⟨character 2⟩}
```

```
\MakeAutoQuote*{⟨character 1⟩}{⟨character 2⟩}
```

Alle Aktivzitate, die mit `\MakeAutoQuote` definiert werden, funktionieren wie das `\enquote`. Die mit `\MakeOuterQuote` und `\MakeInnerQuote` definierten Zitate besitzen nur einen Teil deren Funktionalität. `\MakeAutoQuote*` ist analog zu `\MakeInnerQuote`, d. h. es korrespondiert zu `\enquote*`. Das erstere entspricht dem äußeren Level von `\enquote` wohingegen das letztere der mit Stern versehen Version entspricht. `\MakeAutoQuote*` ist analog zu `\MakeInnerQuote`, d. h. es entspricht `\enquote*`.

## 4.2 Fremdsprachige Zitate

Dieser Befehl verbindet `\MakeAutoQuote` mit dem Sprachwechsel der `babel` oder `polyglossia` Pakete:

```
\MakeForeignQuote{⟨lang⟩}{⟨character 1⟩}{⟨character 2⟩}
```

```
\MakeForeignQuote*{⟨lang⟩}{⟨character 1⟩}{⟨character 2⟩}
```

Die Aktivzitate, die mit den oben stehenden Befehlen definiert werden, entsprechen in ihren Befehlen und der Funktionsweise `\foreignquote` und `\foreignquote*`.

```
\MakeHyphenQuote{⟨lang⟩}{⟨character 1⟩}{⟨character 2⟩}
```

```
\MakeHyphenQuote*{⟨lang⟩}{⟨character 1⟩}{⟨character 2⟩}
```

Die Aktivzitate, die mit den oben stehenden Befehlen definiert werden, entsprechen in ihren Befehlen und der Funktionsweise `\hyphenquote` und `\hyphenquote*`.

## 4.3 Blockzitate von normalem Text

`\MakeBlockQuote` definiert Aktivzitate, setzt dann Zitate im Inline-Modus oder als einen separaten Absatz, abhängig von deren Länge. Dieser Befehl benötigt drei verschiedene (eindeutige) Argumente:

`\MakeBlockQuote{⟨character 1⟩}{⟨delimiter⟩}{⟨character 2⟩}`

Die Argumente werden auf Richtigkeit überprüft, siehe § 10.3, um mehr darüber zu erfahren. Alle mit `\MakeBlockQuote` definierten Aktivzitate verhalten sich dem Wesen nach genauso wie `\blockquote`, jedoch die Handhabung des Zitats unterscheidet sich ein wenig. `⟨character 1⟩` legt die Anfangsmarkierung in der Quelldatei fest und `⟨character 2⟩` die Endmarkierung. Das Symbol, das vom mittleren Argument `⟨delimiter⟩` festgelegt wird, fungiert als Platzhalter, welcher das Zitat von dem Text separiert, aus dem es stammt, wenn Aktivzitate verwendet werden:

```
\MakeBlockQuote{<|>}{>}
...
<text|citation>
```

Wenn das Trennzeichen ausgelassen wird, so wird mit dem Text zwischen der Beginn- und Schlussmarke so umgegangen, als wäre alles zitierter Text.

#### 4.4 Fremdsprachige Blockzitate

Diese Befehle bündeln `\MakeBlockQuote` und die Sprachauswahlfunktion vom `babel`- oder `polyglossia`-Paket:

`\MakeForeignBlockQuote{⟨lang⟩}{⟨character 1⟩}{⟨delimiter⟩}{⟨character 2⟩}`

Die Aktivzitate, die mit diesen Befehlen definiert werden, sind identisch in Funktion und Konzept mit `\foreignblockquote`. Die Verwendung des Platzhaltersymbols funktioniert wie mit `\MakeBlockQuote`.

`\MakeHyphenBlockQuote{⟨lang⟩}{⟨character 1⟩}{⟨delimiter⟩}{⟨character 2⟩}`

Die Aktivzitate, die mit diesen Befehlen definiert werden, sind identisch in Funktion und Konzept mit `\hyphenblockquote`. Die Verwendung des Platzhaltersymbols funktioniert wie mit `\MakeBlockQuote`.

`\MakeHybridBlockQuote{⟨lang⟩}{⟨character 1⟩}{⟨delimiter⟩}{⟨character 2⟩}`

Die Aktivzitate, die mit diesen Befehlen definiert werden, sind identisch in Funktion und Konzept mit `\hybridblockquote`. Die Verwendung des Platzhaltersymbols funktioniert wie mit `\MakeBlockQuote`.

#### 4.5 Aktivzitate überwachen

Die oben eingeführten Befehle können nur Aktivzitate zugewiesen werden, jedoch sind diese Symbole nicht sofort aktiv. Alle zugewiesenen Zitate sind automatisch

ab dem Beginn des Dokumentenkörpers aktiviert. Sollten Aktivzitate im Dokumentenkörper zugewiesen werden, so müssen sie erst durch `\EnableQuotes` aktiviert werden. Die folgenden Befehle steuern den Zustand der Aktivzitate innerhalb eines lokalen Geltungsbereiches.

- `\EnableQuotes` Es aktiviert alle Aktivzitate durch die Neudefinierung der zugewiesenen Symbole und deren Aktivierung. Es stellt sie auch dann auf aktiv um, wenn sie vorher deaktiviert, im Verbatimmodus sind oder überschrieben wurden.
- `\DisableQuotes` Es deaktiviert alle Aktivzitate, indem es die Original-Codeversionen und Definitionen aller zugewiesenen Symbole wiederherstellt.
- `\VerbatimQuotes` Es steuert den Wechsel zur Verbatim-Aktivitäten. Alle Aktivzitate werden dadurch wortwörtlich ausgegeben, solange das Standardverhalten mit `\EnableQuotes` wieder hergestellt werden kann.
- `\DeleteQuotes` Es deaktiviert und gibt alle Aktivzitate frei, d. h. es wird ein kompletter RESET aller zugewiesenen Symbole vorgenommen, sodass diese nun neu definiert werden könnten.

## 5 Integrierte Schnittstellen

Die in diesem Kapitel vorgestellten Befehle sind erweiterte Versionen von einigen in § 3 vorgestellten. Sie unterscheiden sich von ihren Gegenstücken dadurch, dass sie in ihre Syntax automatische Zitate integrieren. Anstelle das man `\cite` manuell hinzufügt, muss man nun nur noch die Zitatargumente der jeweiligen Zitat-Befehle angeben. Siehe § 8.6, wenn Sie einen anderen Befehl als `\cite` verwenden möchten, um die Zitate zu handhaben.

### 5.1 Formelle Zitate von Normaltext

Der Standard-Befehl ist eine erweiterte Version von `\textquote`:

```
\textquote [prenote] [postnote] {key} [punct] {text} tpunct  
\textcquote* [prenote] [postnote] {key} [punct] {text} tpunct
```

Der `<text>` kann jedes beliebige durch Anführungszeichen eingegrenzte Textstück sein. Die optionalen Argumente `<cite>` und `<punct>` spezifizieren die Zitatvariante und die terminalen Satzzeichen des `<text>`es. `<tpunct>` kennzeichnet die dem Befehl folgende Zeichensetzung. Wenn die `autopunct`-Option aktiviert wurde, reguliert diese, ob die Zitierungsbefehle vorausschauend auf Satzzeichen reagieren und diese, wenn nötig, bewegen. Um zu erfahren, wie diese Argumente gehandhabt werden, siehe § 8.7 und siehe § 9.2, um herauszufinden, warum Sie die Zeichensetzung als

separates Argument verändern könnten. Die Version mit Stern des Befehls springt direkt zur inneren Zitatebene. Die übrigen Argumente werden an `\cite` übergeben. Man bemerke, dass `\cite` normalerweise nur die optionalen Argumente unterstützt. `\prenote` zu verwenden ist nur dann möglich, wenn gleichzeitig das `natbib`-, `jurabib`- oder `biblatex`-Paket verwendet wird. Wie diese Argumente gehandhabt werden, hängt vom Zitatbefehl ab. Mit `natbib` und `biblatex` ist `\prenote` wie eine Notiz ‚see‘. Wird hingegen `jurabib` verwendet, so bekommt dieses Argument eine andere Funktion. Das `\postnote` Argument, dessen Einsatz stets möglich ist, erzeugt eine dem Zitat angehängte Notiz, gewöhnlicherweise eine Seitenziffer. `\key` ist der Zitatschlüssel. Siehe §§ 8.6 and 8.7, um herauszufinden, wie Sie ihr Zitat individuellen Wünschen anpassen.

## 5.2 Formelle Zitate in einer Fremdsprache

Die folgenden Befehle verbinden `\textquote` mit der Sprachwechselfunktion der `babel` oder `polyglossia` Pakete:

```
\foreigntextquote{<lang>}[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
\foreigntextquote*{<lang>}[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
```

Dieser Befehl verbindet `\textquote` mit `\foreignlanguage`. Die Argumente werden wie bei `\textquote` behandelt.

```
\hyphentextquote{<lang>}[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
\hyphentextquote*{<lang>}[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
```

Dieser Befehl verbindet `\textquote` mit der `hyphenrules` Umgebung. Die Argumente werden wie bei `\textquote` behandelt.

## 5.3 Blockzitate von regulärem Text

Blockzitate können mit automatischen Zitaten unter der Verwendung der erweiterten Version von `\blockquote` kombiniert werden:

```
\blockcquote[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
```

Der Unterschied zwischen `\blockcquote` und `\blockquote` besteht darin, dass das Eine drei Argumente verwendet anstelle von einem. Die Argumente werden wie bei `\textquote` behandelt; siehe § 5.1 für genauere Informationen. Siehe auch §§ 8.6, 8.7, 9.2 um herauszufinden, wie Sie ihr Blockzitat individuellen Wünschen anpassen.

## 5.4 Blockzitate in einer Fremdsprache

Die folgenden Befehle verbinden `\blockquote` mit der Sprachwechselfunktion der `babel` oder `polyglossia` Pakete:

```
\foreignblockquote{<lang>}[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
```

Dieser Befehl kombiniert `\blockquote` mit `\foreignlanguage`. Lange Zitate werden durch die `otherlanguage*` Umgebung eingeschlossen. Die Argumente werden wie bei `\textquote` behandelt.

```
\hyphenblockquote{<lang>}[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
```

Dieser Befehl verbindet `\blockquote` mit der `hyphenrules` Umgebung. Die Handhabung der Zitatargumente entspricht denen von `\textquote`.

```
\hybridblockquote{<lang>}[<prenote>][<postnote>]{<key>}[<punct>]{<text>}<tpunct>
```

Dieser Befehl arbeitet für kurze Zitate genau wie `\hyphenblockquote` und für lange wie `\foreignblockquote`. Die Handhabung der Zitatargumente entspricht denen von `\textquote`.

## 6 Darstellungsumgebungen

Die Umgebungen, die in diesem Kapitel eingeführt werden, setzen das Zitat als separaten Absatz, der dann genauso aussieht wie ein langes Zitat, das mit den Blockzitanwendungen gesetzt wurde. Dies ist für Zitate vorgesehen, die in einem eigenständigem Absatz (unabhängig der Länge des Zitates) stehen sollen. Man beachte, dass diese Umgebung die Standard-`quote`-Umgebung im nachhaltigem Sinne nicht ersetzen kann. Sie funktionieren wie eine zusätzliche Schicht auf der Oberseite der anderen, wie bei den Blockzitanwendungen. Der Vorteil der Nutzung dieser Umgebung gegenüber dem Standardverfahren, `quote`, ist, dass es konfigurierbar ist, es Zitierungen erlaubt und es immer mit den Blockzitanwendungen synchronisiert, wobei die vorkonfigurierten Optionen, die in den §§ 8.6 and 8.7 diskutiert wurden, beachtet werden.

### 6.1 Grundlegende Darstellungsumgebungen

Die Argumente aller Darstellungsumgebungen sind für gewöhnlich von dem `\begin-` Teil der Umgebung anhängend:

```
\begin{displayquote}[\langle cite \rangle][\langle punct \rangle]
\end{displayquote}
```

Die optionalen Argumente `\langle cite \rangle` und `\langle punct \rangle` spezifizieren die Zitierung und die begrenzende Zeichensetzung des Zitates. In den §§ 8.6 and 8.7 wird näher erläutert, wie man diese Umgebung anpasst. In § 8.7 wird auch beschrieben, wie man die optionalen Argumente noch verwenden kann und in § 9.2 kann man erfahren, warum es gut ist, die begrenzende Zeichensetzung des Zitates in einem eigenen Argument zu separieren. Nachlaufender weißer Platz am Ende der Umgebung wird automatisch entfernt. Es gibt zwei zusätzliche Umgebungen, die `displayquote` mit dem Sprachwechsel des `babel` oder `polyglossia` Paketes verbindet:

```
\begin{foreigndisplayquote}{\langle lang \rangle}[\langle cite \rangle][\langle punct \rangle]
\end{foreigndisplayquote}
```

Diese Umgebung verbindet `displayquote` mit `otherlanguage*`. Abgesehen von der Sprache werden die Argumente wie bei `displayquote` verwendet.

```
\begin{hyphendisplayquote}{\langle lang \rangle}[\langle cite \rangle][\langle punct \rangle]
\end{hyphendisplayquote}
```

Diese Umgebung verbindet `displayquote` mit `hyphenrules`. Abgesehen von der Sprache werden die Argumente wie bei `displayquote` verwendet.

## 6.2 Integrierte Darstellungsumgebungen

Die folgende Umgebung ist eine erweiterte Version von `displayquote`: `displaycquote`:

```
\begin{displaycquote}[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}[\langle punct \rangle]
\end{displaycquote}
```

Der Unterschied zwischen `displaycquote` und dessen Standardgegenstück besteht in den drei Argumenten für die Zitierung anstatt nur einem Argument. Die Platzierung des Zitates ist identisch zu `displayquote` und die Handhabung der Argumente ist wie bei `\textcquote`, das in § 5.1 eingeführt wurde. In §§ 8.6 and 8.7 erfahren Sie, wie diese Umgebung angepasst werden kann. Außerdem wird in § 8.7 darauf eingegangen, wie man mit den optionalen Argumenten umgeht und in § 9.2 wird darauf eingegangen, warum es sinnvoll ist, ein eigenes Argument für die begrenzende Zeichensetzung des Zitates separat einzuführen. Es gibt zwei Umgebungen die `displaycquote` mit dem `babel`- oder `polyglossia`-Paket kombinieren:

```
\begin{foreigndisplayquote}{\langle lang \rangle} [\langle prenote \rangle] [\langle postnote \rangle] {\langle key \rangle} [\langle punct \rangle]
\end{foreigndisplayquote}
```

Diese Umgebung verbindet `displayquote` mit `otherlanguage*`. Abgesehen von der Sprache werden die Argumente wie bei `displayquote` verwendet.

```
\begin{hyphendisplayquote}{\langle lang \rangle} [\langle prenote \rangle] [\langle postnote \rangle] {\langle key \rangle} [\langle punct \rangle]
\end{hyphendisplayquote}
```

Diese Umgebung verbindet `displayquote` mit `hyphenrules`. Abgesehen von der Sprache werden die Argumente wie bei `displayquote` gehandhabt.

## 7 Hilfskommentare

Bei zitiertem formalen Text, bei dem alle Änderungen des zitierten Materials vorgenommen wurden, wie bei Auslassungen, Insertionen oder Alterationen, die typischerweise durch Auslassungspunkte und Anführungszeichen oder durch Klammern gekennzeichnet sind, nutzen Sie die folgenden Befehle, um solche Änderungen in einem formalen Zitat zu kennzeichnen:

```
\textelp{}
\textelp{\langle text \rangle}
\textelp*{\langle text \rangle}
```

Bei Benutzung eines leeren `\langle text \rangle`-Arguments, gibt dieser Befehl Auslassungspunkte aus, um anzuzeigen, dass im Zitat Material ausgelassen wurde. Wenn man ein nicht-leeres Argument verwendet hat, wird den Auslassungspunkten noch ein `\langle text \rangle` in umschlossenen Anführungszeichen angehängt, um anzuzeigen, dass das `\langle text \rangle` nach dem ausgelassenem Material eingefügt wurde. Die mit einem Stern versehene Variante kehrt die Reihenfolge um, d. h. es gibt `\langle text \rangle` gefolgt von Auslassungspunkten aus, um anzuzeigen, dass das `\langle text \rangle` vor dem ausgelassenen Material eingefügt wurde. Insgesamt gibt es drei Wege um diesen Befehl zu benutzen:

```
\textelp{} = [...]
\textelp{\langle text \rangle} = [...] [\langle text \rangle]
\textelp*{\langle text \rangle} = [\langle text \rangle] [...]
```

Die Insertion von Text oder individuellen Buchstaben kann mit dem folgenden Befehl durchgeführt werden:

Zitatdesign	Designvariante
croatian (Kroatisch)	quotes, guillemets, guillemets*
danish (Dänisch)	quotes, guillemets
dutch (Holländisch)	–
english (Englisch)	american, british
finnish (Finnisch)	–
french (Französisch)	quotes, quotes*, guillemets, guillemets*
german (Deutsch)	quotes, guillemets, swiss
greek (Griechisch)	–
italian (Italienisch)	quotes, guillemets
norwegian (Norwegisch)	guillemets, quotes
portuguese (Portugiesisch)	portuguese, brazilian
russian (Russisch)	–
spanish (Spanisch)	spanish, mexican
swedish (Schwedisch)	quotes, guillemets, guillemets*

Tabelle 2: Zitatdesigns und Designvarianten vordefiniert

```
\textins{<text>}
\textins*{<text>}
```

Standardmäßig wird `\textins` das `<text>` umschließen, das dem zitierten Material in Anführungszeichen hinzugefügt wurde. Die Sternvariante ist für kleinere Änderungen da, wie die Großschreibung eines Wortes, die erforderlich sind, um das zitierte Material an den neuen Kontext, in dem es zitiert wurde, anzupassen.

```
\textins{text} = [text]
\textins*{T}ext = [T]ext
```

Siehe § 8.10, um herauszufinden, wie man das Aussehen von Auslassungspunkten und Insertionen konfiguriert.

## 8 Konfiguration

Nach Möglichkeit lädt dieses Paket die Konfigurationsdatei `csquotes.cfg`. Man kann diese Datei verwenden, um neue Zitatdesigns oder Synonyme hinzuzufügen oder bereits existierende neu zu belegen.

### 8.1 Zitatdesigns definieren

Benutze den folgenden Befehl, um Zitatdesigns oder Designvarianten zu definieren:

```
\DeclareQuoteStyle[<variant>]{<style>}[<outer init>][<inner init>]%
```

```
{⟨opening outer mark⟩}[⟨middle outer mark⟩]{⟨closing outer mark⟩}[⟨kern⟩]%
{⟨opening inner mark⟩}[⟨middle inner mark⟩]{⟨closing inner mark⟩}
```

Dieser Befehl kann in der Präambel oder in der Konfigurierungsdatei verwendet werden. Der Bezeichner ‚outer‘ bezieht sich auf die erste Zitatebene und ‚inner‘ meint ein Zitat aus einem anderen Zitat. Eine ‚middle mark‘ ist eine Zitatmarkierung am Anfang eines jeden Absatzes, sollte das Zitat sich über mehrere Absätze erstrecken. In den meisten Fällen handelt es sich bei den Anführungszeichen definierenden Argumenten, um eines der in Tabelle 6 aufgelisteten. Wenn sowohl ein inneres als auch ein äußeres Zitat an der selben Stelle anfängt oder endet, wird der Abstand der zwischen den benachbarten Anführungszeichen eingefügt, wird angegeben durch den *⟨kern⟩* Wert. Weil dieser Wert durch jede in TeX bekannte Einheit angegeben werden kann, ist es gewöhnlich vorgeschrieben eine relationale font-dependent Einheit ‚em‘ anstelle von absoluten Einheiten wie z. B. Points, Inches oder Millimeter zu verwenden. Man bemerke, dass *⟨kern⟩* nur als Rückgriffwert verwendet wird. Wenn die Schriftart eine Kerningdatei für das entsprechende Paar von Anführungszeichen besitzt, wird das Abstandsmaß der Kerningdatei vorrangig verwendet. *⟨outer init⟩* und *⟨inner init⟩* sind Universalwerkzeuge für die Initialisierung eines Zitatdesigns. Durch die Auswahl eines Zitatdesigns werden diese Werkzeuge für alle Zitatbefehle verfügbar, ohne diese erweitern zu müssen. Das Ausführen von *⟨outer init⟩* findet sofort, noch bevor das äußere Zitat eingefügt wird, statt. Zwischen diesem Paar befindet sich das Zitat. *⟨inner init⟩* wird ausgeführt, bevor der Anfang des inneren Zitates eingefügt wurde. Es empfiehlt sich in diesem Rahmen alle globalen Zuweisungen zu vermeiden, um Störungen mit anderen Designs zu verhindern. Immer, wenn es ein *⟨inner init⟩* gibt, muss auch ein *⟨outer init⟩* definiert werden, auch dann wenn dieses Argument leer ist. In Tabelle 2 finden Sie eine Aufzählung aller vordefinierten Zitatdesigns und deren Varianten. Das sind ausschließlich unterstützte Designs, siehe auch Tabelle 3 eine Aufzählung aller Sprachsynonyme. Siehe § 9.1 für einige Beispiele und eine Übersicht, die darstellt wie Zitatdesigns, Synonyme und Paketoptionen zusammenarbeiten.

## 8.2 Zitatsynonyme Definieren

Der folgende Befehl definiert Zitatsynonyme:

```
\DeclareQuoteAlias[⟨variant⟩]{⟨style⟩}{⟨alias⟩}
\DeclareQuoteAlias{⟨first-level alias⟩}{⟨second-level alias⟩}
```

Dieser Befehl kann sowohl in der Konfigurationsdatei als auch in der Präambel des Dokuments verwendet werden. Die Synonyme müssen entweder auf eines der bereits existierenden Designs oder auf ein anders Synonym verweisen. Die meisten

Synonym	Backenddesign / Synonym	Synonym	Backenddesign / Synonym
american	english/american	naustrian	austrian
australian	english/british	newzealand	english/british
austrian	german/quotes	ngerman	german
brazil	brazilian	norsk	norwegian
brazilian	portuguese/brazilian	norwegian	norwegian/guillemets
british	english/british	nynorsk	norwegian
canadian	english/american	portuges	portuguese
croatian	croatian/quotes	portuguese	portuguese/portuguese
danish	danish/quotes	spanish	spanish/spanish
english	english/american	swedish	swedish/quotes
french	french/quotes	swiss	german/swiss
german	german/quotes	UKenglish	british
italian	italian/quotes	USEnglish	american
mexican	spanish/mexican		

Tabelle 3: Voreingestellte Sprachsynonyme

Synonyme für Sprachen verweisen auf ein bereits existierendes Design, doch es existieren auch einige die erst auf ein intermediates Synonym verweisen. Wenn das Synonym für eines der beiden Pakete `babel` oder `polyglossia` definiert wurde, so muss dessen Name identisch sein zu den in den Paketen verwendeten Sprachnamen. D. h. sie müssen Erweiterungen von `\language` sein. Siehe § 9.1 für eine Übersicht, die darstellt, wie Zitatdesigns, Synonyme und Paketoptionen zusammenhängen. Eine Aufzählung aller voreingestellten Synonyme finden Sie in Tabelle 3.

### 8.3 Paketoptionen definieren

Der folgende Befehl erzeugt eine neue Paketoption, basierend auf einer Schlüssel/Wert-Syntax. Es wird der Designname als zwingend erforderliches Argument benötigt:

```
\DeclareQuoteOption{<style>}
```

Wenn Sie eine neue Option verwenden, wird der Name des Zitatdesigns als Schlüssel fungieren. Der Wert kann jede Designvariante des entsprechenden Designs sein. Die Paketoption wählt eine Variante, indem sie ein Synonym definiert, das auf das entsprechende Design verweist. Diesen Befehl können Sie nur in der Konfigurationsdatei verwenden. Siehe § 9.1 für eine Übersicht die darstellt, wie Zitatdesigns, Synonyme und Paketoptionen zusammenhängen.



```
\SetBlockThreshold{<integer>}
\SetBlockEnvironment{<environment>}
\SetCiteCommand{<command>}
```

`\SetBlockThreshold` legt die Anzahl von Zeilen oder Wörtern fest, die die Blockzitanwendung als Schwellenwert verwendet, um festzulegen, ab wann ein Zitat im Inline- oder im Display-Modus gesetzt werden soll. Voreingestellt ist drei. `\SetBlockEnvironment` gibt die verwendete Umgebung, die für Zitate im Block- oder Display-Modus verwendet wird, an. Es verwendet den Namen einer existierenden Umgebung als Argument. Voreingestellt ist, dass die `quote`-Umgebung, die für die meisten Dokumentklassen vorgesehen ist. Das Argument von `\SetCiteCommand` gibt an, welcher Ersatz für `\cite` durch die integrierten Zitanwendungen verwendet wird, um die Zitate zu behandeln. Es muss ein einzelner Befehl sein, der ein oder zwei optionale Argumente gefolgt von einem notwendigen Argument verwendet; das Notwendige fungiert als Zitatschlüssel. Voreingestellt ist `\cite`. Die Zitatbefehle der `natbib`, `jurabib`, und `biblatex` Pakete, die zwei optionale Argumente verwenden, werden unterstützt.

## 8.7 Haken für Anführungszeichen und Zitate

Das Aussehen von Anführungszeichen kann in geringem Maße durch Neudefinierung des Haken konfiguriert werden, was Ihnen unten erklärt wird. Dieses Kapitel gibt einen Überblick über die Syntax von Haken. Siehe § 9.2 für praxisnahe Beispiele. Die Zitanwendungen, mit denen diese Haken verwendet werden können, sind in Tabelle 5 aufgelistet. Siehe auch § 8.8 für Tests, die sinnvoll sein können, wenn Sie Haken neu definieren.

```
\mkcitation{<cite>}
```

Alle Anwendungen, die das `<cite>`-Argument verwenden, geben dieses weiter an den `\mkcitation` Haken, der neu definiert wird, um das Zitat aufzubereiten. `\mkcitation` wird nur dann angewendet, wenn auch ein Zitat vorhanden ist. Voreingestellt separiert es das Zitat von dem vorangehenden Text durch einen Interwort-Raum und schließt es in Klammern ein. Das ist analog zur folgenden Definition:

```
\newcommand*{\mkcitation}[1]{\_#1}
```

```
\mkccitation{<cite code>}
```

Die integrierten Zitanwendungen verwenden `\mkccitation` anstelle von `\mkcitation`. Der Befehl separiert das Zitat im voreingestellten Modus vom vorangestellten Text durch einen Interwort-Raum. Das ist analog zur folgenden Definition:

```
\newcommand*{\mkccitation}[1]{\_#1}
```

Haken	Befehl oder Umgebung																				
	<code>\enquote</code>	<code>\foreignquote</code>	<code>\hyphenquote</code>	<code>\textquote</code>	<code>\foreigntextquote</code>	<code>\hyphenntextquote</code>	<code>\textcquote</code>	<code>\foreigntextcquote</code>	<code>\hyphenntextcquote</code>	<code>\blockquote</code>	<code>\foreignblockquote</code>	<code>\hyphenblockquote</code>	<code>\blockcquote</code>	<code>\foreignblockcquote</code>	<code>\hyphenblockcquote</code>	<code>displayquote</code>	<code>foreigndisplayquote</code>	<code>hyphendisplayquote</code>	<code>displaycquote</code>	<code>foreigndisplaycquote</code>	<code>hyphendisplaycquote</code>
<code>\mktextquote</code>	-	-	-	•	•	•	•	•	•	•	•	•	•	•	•	-	-	-	-	-	-
<code>\mkblockquote</code>	-	-	-	-	-	-	-	-	-	•	•	•	•	•	•	-	-	-	-	-	-
<code>\mkbegdispquote</code>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	•	•	•	•	•	•
<code>\mkenddispquote</code>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	•	•	•	•	•	•
<code>\mkcitation</code>	-	-	-	•	•	•	-	-	-	•	•	•	-	-	-	•	•	•	-	-	-
<code>\mkccitation</code>	-	-	-	-	-	-	•	•	•	-	-	-	•	•	•	-	-	-	•	•	•

Tabelle 5: Vorhandensein von Zusatzhaken

`\mktextquote`{*open*}{*text*}{*close*}{*punct*}{*tpunct*}{*cite*}

Der `\mktextquote`-Haken steuert das Aussehen von allen Textzitaten. Dieser Haken wird von `\textquote` und allen damit verwandten Befehlen aus §§ 3.3, 3.4, 5.1, 5.2 verwendet. `\blockquote` und dessen Verwandte aus §§ 3.5, 3.6, 5.3, 5.4 verwenden diesen Haken für kurze Zitate. Er braucht sechs Argumente, die entsprechend der gewünschten Ausgabe angeordnet werden können:

- #1 Das öffnende Anführungszeichen.
- #2 Das *text*-Argument des Befehls.
- #3 Das abschließende Anführungszeichen.
- #4 Das optionale *punct*-Argument des Befehls. Wenn dort kein *punct* Argument ist, wird dieser Parameter leer gelassen.
- #5 Anhängende *tpunct*-Zeichensetzung direkt nach dem Befehl. Wenn dort keine solche Zeichensetzung ist oder das `autopunct`-Feature deaktiviert ist, ist der Parameter leer.
- #6 Das optionale *cite*-Argument des Befehls ist verpackt in `\mkcitation`. Wenn dort kein `\cite`-Argument ist, bleibt auch dieser Parameter leer. Mit integrierten Zitatbefehlen enthält dieser Parameter Zitatcode in `\mkccitation` verpackt.

Standardmäßig umschließt `\mktextquote` das *punct* Argument in Anführungszeichen zusammen mit dem *text* und fügt das *cite* Argument oder den Zitatcode vor jeglicher anhängender *tpunct* Zeichensetzung ein. Das ist analog zur folgenden Definition:

```
\newcommand{\mktextquote}[6]{#1#2#4#3#6#5}
```

Wie `\mktextquote` sich in den Formatierungsprozess einhakt, kann man am besten an einem Beispiel sehen. Die Befehle

```
\textquote[cite]{short quote}
\textcquote[55]{key1}[.]{short quote}
\blockcquote[87]{key2}{short quote}.
```

wenden `\mktextquote` mit den folgenden Argumenten an:

```
\mktextquote{open}{short quote}{close}{ }{\mkcitation{cite}}
\mktextquote{open}{short quote}{close}{. }{\mkccitation{\cite[55]{key1}}}
\mktextquote{open}{short quote}{close}{.}{\mkccitation{\cite[87]{key2}}}
```

Dort wo `\cite` verwendet wird, wird der Befehl mit `\SetCiteCommand` ausgewählt und `open/close` sind interne Makros, welche die öffnenden und abschließenden Anführungszeichen ausgeben. Achten Sie darauf, dass die internen Makros vollwertige Markup-Elemente sind mit einer Gruppierungs- und Schachtelungskontrolle. Sie müssen in der korrekten Reihenfolge eingesetzt werden, sonst wird `csquotes` Fehler ausgeben, betreffend nicht balancierter Gruppierungen oder unkorrekt geschachtelter Zitate. Da der `<text>` sinnvollerweise in Anführungszeichen eingeschlossen sein sollte, wird die Parameterreihenfolge `#1#2#3` effektiv repariert. Die Parameter `#4`, `#5`, `#6` können frei platziert werden.

```
\mkblockquote{<text>}{<punct>}{<tpunct>}{<cite>}
```

Der `\mkblockquote` Haken reguliert das Aussehen aller Zitatblöcke. Dieser Haken wird von `\blockquote` und seinen verwandten Befehlen aus §§ 3.5, 3.6, 5.3, 5.4 für lange Zitate verwendet. Er benötigt vier Argumente, die entsprechend der gewünschten Ausgabe angeordnet werden können:

- #1 Das `<text>`-Argument des Befehls.
- #2 Das optionale `<punct>`-Argument des Befehls. Wenn dort kein `<punct>` Argument ist, wird dieser Parameter leer gelassen.
- #3 Anhängende `<tpunct>`-Zeichensetzung direkt nach dem Befehl. Wenn dort keine solche Zeichensetzung ist oder das `autopunct`-Feature deaktiviert ist, ist der Parameter leer.
- #4 Das optionale `<cite>`-Argument des Befehls, verpackt in `\mkcitation`. Wenn dort kein `\cite`-Argument ist, bleibt auch dieser Parameter leer. Mit integrierten Zitatbefehlen enthält dieser Parameter Zitatcode in `\mkccitation` verpackt.

Standardmäßig fügt `\mkblockquote` das `<cite>`-Argument oder den Zitatcode direkt hinter dem `<text>` ein und fügt die anhängende `<punct>` Zeichensetzung an das Ende an. Das ist analog zur folgenden Definition:

```
\newcommand{\mkblockquote}[4]{#1#2#4#3}
```

```
\mkbegdispquote{<punct>}{<cite>}
```

```
\mkenddispquote{<punct>}{<cite>}
```

Die `\mkbegdispquote` und `\mkenddispquote` Haken werden von `displayquote` und dessen verwandten Umgebungen aus §§ 6.1 and 6.2 verwendet. Diese Haken benötigen zwei Argumente:

- #1 Das `<punct>`-Argument wird an `\begin` Zeile der Umgebung übergeben. Wenn dort kein `<punct>`-Argument ist, wird dieser Parameter leer gelassen.
- #2 Das `<cite>`-Argument wird verpackt in `\mkcitation` und an die Umgebung übergeben. Wenn dort kein `\cite`-Argument ist, bleibt auch dieser Parameter leer. Mit intergierten Zitatbefehlen enthält dieser Parameter Zitatcode in `\mkccitation` verpackt.

Standardmäßig fügt `\mkenddispquote` das `<punct>` Argument sowie das `<cite>` Argument oder den Zitatcode am Ende des Zitates ein. `\mkbegdispquote` fügt standardmäßig nichts ein. Das ist analog zu der folgenden Definition:

```
\newcommand{\mkbegdispquote}[2]{}
```

```
\newcommand{\mkenddispquote}[2]{#1#2}
```

Siehe § 9.2 für praktische Beispiele.

## 8.8 Zusätzliche Tests für Zitathaken

Die Befehle in diesem Kapitel erhöhen die Flexibilität der in § 8.7 vorgestellten Haken. Zum Beispiel wäre es doch wünschenswert, das Format des Zitates so einzustellen, dass es zur Setzung des korrespondierenden Zitates passt. Es wäre auch toll zu wissen, ob das Zitat mit einem Satzzeichen endet.

```
\ifpunctmark{<character>}{<>true>}{<>false>}
```

Es wird `<>true>` ausgegeben, wenn es vor einem Satzzeichen `<character>` steht und `<>false>` in allen anderen Fällen. Der `<character>` (Symbol) kann ein Punkt, Komma, Semikolon, Doppelpunkt, Ausrufezeichen oder Fragezeichen sein. Beachten Sie, dass dieser Test nur in der Definition eines Hakens aus § 8.7 möglich ist.

`\ifpunct{⟨true⟩}{⟨false⟩}`

Es wird `⟨true⟩` ausgegeben, wenn es vor einem Satzzeichen `⟨character⟩` steht und `⟨false⟩` in allen anderen Fällen. Es ist zu beachten, dass dieser Test nur in der Definition eines Hakens aus § 8.7 möglich ist.

`\ifterm{⟨true⟩}{⟨false⟩}`

Es wird `⟨true⟩` ausgegeben, wenn es vor einem terminalen Satzzeichen (Punkt, Ausrufezeichen, Fragezeichen) steht und sonst `⟨false⟩`. Beachten Sie, dass dieser Test nur in der Definition eines Hakens aus § 8.7 möglich ist.

`\iftextpunctmark{⟨text⟩}{⟨character⟩}{⟨true⟩}{⟨false⟩}`

Es wird `⟨true⟩` ausgegeben, wenn der `⟨text⟩` mit einem Satzzeichen `⟨character⟩` endet und `⟨false⟩` in allen anderen Fällen. Der `⟨character⟩` (Symbol) kann ein Punkt, Komma, Semikolon, Doppelpunkt, Ausrufezeichen oder Fragezeichen sein. Dieser Befehl ist stabil.

`\iftextpunct{⟨text⟩}{⟨true⟩}{⟨false⟩}`

Es wird `⟨true⟩` ausgegeben, wenn der `⟨text⟩` mit irgendeinem Satzzeichen endet und `⟨false⟩` in allen anderen Fällen. Dieser Befehl ist stabil.

`\iftextterm{⟨text⟩}{⟨true⟩}{⟨false⟩}`

Es wird `⟨true⟩` ausgegeben, wenn es vor einem terminalen Satzzeichen (Punkt, Ausrufezeichen, Fragezeichen) steht und sonst `⟨false⟩`. Der Befehl ist stabil.

`\ifblockquote{⟨true⟩}{⟨false⟩}`

Es wird `⟨true⟩` in allen Block- und Displayzitaten ausgegeben und sonst `⟨false⟩`.

`\ifblank{⟨string⟩}{⟨true⟩}{⟨false⟩}`

Dieser übergeordnete Befehl, der vom `etoolbox`-Paket bereitgestellt wird, gibt `⟨true⟩` aus, wenn der `⟨string⟩` blank (leer oder Freiräume habend) ist und sonst `⟨false⟩`. Das ist hilfreich, um auf leere Argumente in den Definitionen der `\mk...quote` Befehle zu testen. Beachte, dass dieser Test redundant wäre in der Definition von Zitat-Haken, da diese nur ausgeführt werden, wenn überhaupt ein Zitat zur Verfügung steht.

`\unspace` Entfernt vorangehenden Leerraum, d. h. er entfernt alle Sprünge und die Strafen aus dem Ende der aktuellen horizontalen Liste.

## 8.9 Vorausschauendes Konfigurieren der Interpunktion

```
\DeclareAutoPunct{⟨characters⟩}
```

Dieser Befehl definiert das Satzzeichen, das von Zitatbefehlen verwendet wird, indem er vorausschauend nach Satzzeichen sucht. Beachten Sie, dass *⟨characters⟩* eine unendliche Liste von Symbolen ist. Mögliche *⟨characters⟩* wären Punkt, Komma, Semikolon, Doppelpunkt, Ausrufe- und Fragezeichen. Die Standardeinstellung ist:

```
\DeclareAutoPunct{.,;:!?}
```

Diese Definition ist automatisch wieder eingestellt, wenn die `autopunct` Paketoption auf `true` gesetzt wird. Wenn man `\DeclareAutoPunctuation{}` anwendet, ist das analog zu `autopunct=false`, d. h. es deaktiviert dieses Feature.

## 8.10 Auslassungszeichen konfigurieren

Das Aussehen von Auslassungszeichen und Insertionen, reguliert durch die Hilfsbefehle aus § 7, wird durch fünf Haken gesteuert. Wenn `\textelp` mit einem leeren Argument verwendet wird (nur Auslassungen), wendet es `\mktextelp` an. Wenn ein nicht-leeres Argument verwendet wird (Auslassung und Insertion), wird der *⟨text⟩* als Argument an `\mktextelpins` weitergegeben. Die mit einem Stern versehene Version wird den *⟨text⟩* hingegen an `\mktextinselp` übergeben. Das sind Standarddefinitionen:

```
\newcommand{\mktextelp}{[\textellipsis\unkern]}
\newcommand{\mktextelpins}[1]{[\textellipsis\unkern]_{#1}}
\newcommand{\mktextinselp}[1]{[#1]_{[\textellipsis\unkern]}}
```

Der `\textins`-Befehl übergibt sein *⟨text⟩* Argument an `\mktextins` für die weitere Auswertung. Die `\textins` Version mit Stern verwendet dafür stattdessen `\mktextmod`. Das sind Standarddefinitionen:

```
\newcommand{\mktextins}[1]{[#1]}
\newcommand{\mktextmod}[1]{[#1]}
```

Sie können die oben verwendeten Haken neu definieren, um das Format der Ausgabe zu verändern. Zum Beispiel, wenn Sie wollen, dass Ersetzungen durch „[...] [text]“ eher als „[...] [text]“ angezeigt werden sollen, müssen Sie `\mktextelpins` entsprechend neu definieren:

```
\newcommand{\mktextelpins}[1]{[\textellipsis #1]}
```

Das `\unkern` in der Standarddefinition ist erforderlich, weil `\textellipsis` ein asymmetrisches Kernig standardmäßig einfügt. Das Kerning hinter dem letzten Punkt ist analog zu dem Raum zwischen den Punkten, welcher gut ist, solange `\textellipsis` vor irgendwelchem Text steht. Jedoch ist es unerwünscht, wenn `\textellipsis` durch Klammern eingeschlossen ist.

## 9 Hinweise zur Anwendung

### 9.1 Ein neues Zitatdesign hinzufügen

Dieses Kapitel zeigt einige umfassende Beispiele auf, wie neue Zitatdesigns definiert werden. Die hier vorgestellten Beispiele nutzen allerdings nur die grundlegenden Komponenten, aus denen die Designs zusammengesetzt sein können. Der wichtigste Punkt ist, das Zusammenspiel von Zitatdesigns, Varianten, Pseudonymen und Paket-Optionen zu veranschaulichen. Um loslegen zu können, wählen Sie einen einfachen bereits vorhandenen Stil über die Paket-Option `style` oder den Befehl `\setquotestyle`:

```
\DeclareQuoteStyle{house}
  {\textquotedblleft}{\textquotedblright}
  {\textquoteleft}{\textquoteright}
```

Nehmen wir nun an, dass wir ein Zitatdesign für eine imaginäre Sprache namens „Newspeak“ hinzufügen wollen und dass es zwei Arten des Zitatdesigns gibt, die Newspeak üblicherweise verwenden, eine offizielle und eine inoffizielle. In diesem Fall benötigen wir zwei Back-End-Designs, die durch Varianten von `newspeak-Design`, `newspeak/official` und `newspeak/unofficial` realisiert werden:

```
\DeclareQuoteStyle[official]{newspeak}
  {\textquotedblleft}{\textquotedblright}
  {\textquoteleft}{\textquoteright}
\DeclareQuoteStyle[unofficial]{newspeak}
  {\textquotedblright}{\textquotedblleft}
  {\textquoteright}{\textquoteleft}
```

Die offizielle Variante sollte als Standardeinstellung für dieses Design gewählt werden. Es besteht keine Notwendigkeit, die Definition der `official`-Variante auszuführen. Wir definieren einfach eine, mit einem Pseudonym beschriftete, `newspeak`, welche die gewünschte Variante zeigt:

```
\DeclareQuoteAlias[official]{newspeak}{newspeak}
```

Der Grund, warum wir mithilfe von Varianten und Pseudonymen, anstelle von zwei unabhängigen Designs, arbeiten, wird im Folgenden offensichtlich werden. Nehmen wir an, das `babel`-Paket bietet eine Unterstützung von Newspeak an. Aber diese Sprache ist bekanntlich `babel` als `otherspeak`:

```
\DeclareQuoteAlias{newspeak}{otherspeak}
```

Dies ist ein Beispiel für ein zweistufiges Pseudonym, das auf ein einstufiges Pseudonym verweist. Falls die verwendete Sprache `otherspeak` ist, werden die oben genannten Pseudonyme wie folgt erweitert:

```
otherspeak = newspeak = newspeak/official
```

Wir definieren nun eine weitere Paket-Option, um eine Design-Variante zu wählen:

```
\DeclareQuoteOption{newspeak}
```

Dadurch wird eine neue Paket-Option mit dem Schlüssel `newspeak` hinzugefügt. Der Wert dieser Option kann in jeder Variante des `newspeak`-Designs in der Konfigurationsdatei definiert werden. In diesem Beispiel gibt es zwei mögliche Werte: `official` und `unofficial`. Um den Standard- oder Alternativ-Stil für das gesamte Dokument auszuwählen, verwenden wir:

```
\usepackage[style=newspeak]{csquotes}  
\usepackage[style=newspeak,newspeak=unofficial]{csquotes}
```

Um den Standard- oder Alternativ-Stil mit mehrsprachiger Unterstützung auszusuchen, benutzen wir:

```
\usepackage[babel]{csquotes}  
\usepackage[babel,newspeak=unofficial]{csquotes}
```

Das Grunddesign muss in diesem Fall als Pseudonym integriert sein, da die `newspeak`-Option durch Neudefinierung eine Variante auswählt und somit das `newspeak`-Pseudonym überschreibt. Da das `otherspeak`-Pseudonym auf `newspeak` verweist und nicht direkt zu einem Back-End-Design, wird die `newspeak`-Option auch den gewünschten Effekt haben, wenn eine mehrsprachige Unterstützung aktiviert ist. Beachten Sie, dass einige Namen dieses Stils eine besondere Bedeutung haben. Siehe § 10.9 für weitere Details.

## 9.2 Zitate verwenden und Literaturstellen einbinden

Designanleitungen für Autoren geben detaillierte Regeln für die Formatierung von Zitaten und Literaturstellen an, einschließlich Regeln für Zeichensetzungseinstellungen. In diesem Abschnitt werden einige typische Nutzungsszenarien diskutiert, Literaturstelleneinbindung und andere Möglichkeiten wurden in §§ 8.7, 8.8, 8.9 eingeführt. In den nachstehenden Beispielen gehen wir von folgenden Einstellungen aus:

```
\textquote[citation][.]{Dies ist ein vollständiger Satz}  
\textquote[citation][ ]{Dies ist ein unvollständiger Satz}.
```

Wir beginnen mit semantisch strengen Zeichensetzungseinstellungen, d. h. terminale Zeichensetzung steht in Anführungszeichen, falls es tatsächlich Teil des Zitats ist und wird nach dem abschließenden Zeichen gesetzt, wenn es kein Teil des Zitats

ist. Unsere ersten Beispielfälle benutzen die deutschen Anführungszeichen, aber die Formationskonvention ist, soweit sie die Zeichensetzungseinstellungen betrifft, sehr verbreitet. Im ersten Beispiel gehen wir von Literaturstellen in den Fußnoten aus, damit die gewünschte Ausgabe wie folgt lautet:

„Dies ist ein vollständiger Satz.“<sup>1</sup>  
„Dies ist ein unvollständiger Satz“.<sup>2</sup>

Dies wird durch die folgenden Definitionen erreicht:

```
\renewcommand{\mkcitation}[1]{\footnote{#1}}  
\renewcommand{\mktextquote}[6]{#1#2#4#3#5#6}
```

In einigen Fällen ist eine etwas andere Platzierung der Zeichensetzung in der zweiten Zeile gefordert:

„Dies ist ein vollständiger Satz.“<sup>1</sup>  
„Dies ist ein unvollständiger Satz“<sup>2</sup>.

Dies wird durch die folgenden Definitionen erreicht:

```
\renewcommand{\mkcitation}[1]{\footnote{#1}}  
\renewcommand{\mktextquote}[6]{#1#2#4#3#6#5}
```

In den nächsten Beispielen wechseln wir zu amerikanischen Anführungszeichen. Wenn Klammern genutzt werden, wird das terminale Satzzeichen in der Regel ganz ans Ende des ganzen Satzes nach dem schließenden Anführungszeichen und der Literaturangabe gestellt, auch wenn es Teil des ursprünglichen Zitats sind:

“Dies ist ein vollständiger Satz” (Zitat).  
“Dies ist ein unvollständiger Satz” (Zitat).

Dies wird durch die folgenden Definitionen erreicht:

```
\renewcommand{\mkcitation}[1]{ (#1)}  
\renewcommand{\mktextquote}[6]{#1#2#3#6#4#5}
```

Das amerikanische Zitatdesign ist besonders, da sich ein Punkt oder ein Komma unmittelbar nach einem schließenden Anführungszeichen innerhalb des Zitats befinden müssen, auch wenn sie nicht Teil des ursprünglichen Zitats sind. Wenn die oben stehende Eingabe gegeben ist (und unter der Annahme, dass Literaturangaben im nächsten Beispiel in Fußnoten stehen), so benötigen wir in beiden Fällen die gleiche Ausgabe:

“Dies ist ein vollständiger Satz.”<sup>1</sup>

“Dies ist ein unvollständiger Satz.”<sup>2</sup>

Dies wird durch die folgenden Definitionen erreicht:

```
\DeclareAutoPunct{.,}  
\renewcommand{\mkcitation}[1]{\footnote{#1}}  
\renewcommand{\mktextquote}[6]{#1#2#4#5#3#6}
```

Das Design, das gewöhnlich in französischen Büchern genutzt wird, verwendet semantisch strenge Zeichensetzungseinstellungen. Das Besondere an diesem Stil ist das Fußnotenzeichen, welches innerhalb der Anführungszeichen vor dem terminalen Satzzeichen gesetzt wird. Wenn die oben stehende Eingabe gegeben ist, benötigen wir die folgende Ausgabe:

```
« Dies ist ein vollständiger Satz1. »  
« Dies ist ein unvollständiger Satz2 ».
```

Dies wird durch die folgenden Definitionen erreicht:

```
\renewcommand{\mkcitation}[1]{\footnote{#1}}  
\renewcommand{\mktextquote}[6]{#1#2#6#4#3#5}
```

Beachten Sie, dass der Abstand der Anführungszeichen, wie häufig in der französischen Typografie, was beim Zitatdesign behandelt wurde, definiert ist durch `\DeclareQuoteStyle`. Es besteht keine Notwendigkeit, sich mit diesen Details zu befassen, wenn es sich um eine Neudefinition von `\mktextquote` und ähnliche Einbindungen handelt. `\mktextquote` erhält die Anführungszeichen und den Abstand als Parameter #1 and #3. Neben sprach- und designspezifischen Anpassungen wird in diesem Abschnitt diskutiert, wie Formatierungseinbindungen verwendet werden können, um den Zitierungsprozess weiter zu automatisieren. Zum Beispiel kann das automatische Einfügen von Auslassungszeichen konfiguriert werden, wenn das Zitat eines Satzes am Ende abgeschnitten ist. In den folgenden Beispielen gehen wir von diesen Zeilen als Eingabe aus:

```
\textquote[citation]{Dies ist ein unvollständiger Satz}.  
\textquote[citation][.]{Dies ist ein vollständiger Satz}  
\textquote[citation]{Dies ist ein vollständiger Satz.}
```

Nehmen wir an, Zitate amerikanischen Stils werden mit Literaturstellen in der Fußnote kombiniert. Unser bisherige Definition dafür war:

```
\renewcommand{\mktextquote}[6]{#1#2#4#5#3#6}
```

Wir müssen ein Auslassungszeichen einfügen, falls das *(punct)*-Argument des Literaturstellenbefehls leer oder weggefallen ist. Parameter #4 ist in beiden Fällen leer:

```
\renewcommand{\mktextquote}[6]{%
  #1#2\ifblank{#4}{ \textelp{}}{#4}#5#3#6}
```

Wenn die oben genannte Eingabe gegeben ist, wird die Definition folgende Ausgabe liefern:

“Dies ist ein unvollständiger Satz [...]”<sup>1</sup>

“Dies ist ein vollständiger Satz.”<sup>2</sup>

“Dies ist ein vollständiger Satz. [...]”<sup>3</sup>

Die ersten beiden Fälle wurden korrekt behandelt, aber der dritte Fall benötigt mehr Einstellungen: Hier gibt es eine unberechtigte Auslassung, da der letzte Punkt nicht an `\textquote` als separates (*punct*)-Argument übergeben wurde, aber im Zitat enthalten ist. Um dies zu beheben, fügen wir einen weiteren `\ifpunct` Test hinzu, um zu überprüfen, ob der zitierte Text mit einem Satzzeichen endet und die Auslassung wegfällt, wenn das der Fall ist:

```
\renewcommand{\mktextquote}[6]{%
  #1#2\ifblank{#4}{\ifpunct{}{ \textelp{}}}{#4}#5#3#6}
```

Der `\ifpunct`-Test wird das letzte Zeichen in #2 überprüfen und die Auslassung weglassen, wenn ein Satzzeichen erfasst wurde. Daraus ergibt sich die gewünschte Ausgabe:

“Dies ist ein unvollständiger Satz [...]”<sup>1</sup>

“Dies ist ein vollständiger Satz.”<sup>2</sup>

“Dies ist ein vollständiger Satz.”<sup>3</sup>

Bei der Verwendung von automatisierten Literaturstellen, ist es zweckmäßig, integrierte Zitatbefehle aus § 5 anzuwenden. Zum Beispiel anstatt diesen:

```
\textquote[{\cite[55]{key}}][.]{Dies ist ein vollständiger Satz}
\textquote[{\cite[55]{key}}][]{Dies ist ein unvollständiger Satz}.
```

könnten Sie `\textcquote` statt `\textquote` benutzen:

```
\textcquote[55]{key}[.]{Dies ist ein vollständiger Satz}
\textcquote[55]{key}{Dies ist ein unvollständiger Satz}.
```

Die Definition der Zeichensetzungseinbindung ist in beiden Fällen gleich, allerdings werden die Literaturstellen im zweiten Fall auf eine etwas andere Art und Weise gesetzt, es wird nämlich `\mkccitation` anstatt `\mkcitation` verwendet. Die erste Zeile im Beispiel unten ist für Klammerliteraturangaben bestimmt, die zweite für Literaturangaben in Fußnoten:

```
\renewcommand{\mkccitation}[1]{ (#1)}
\renewcommand{\mkccitation}[1]{\footnote{#1}}
```

Erweiterte Literaturstellenpakete wie `natbib` und `biblatex` liefern zugehörige Befehle für verschiedene Arten von Literaturstellen. In diesem Fall ist es ratsam, die Literaturstellen des jeweiligen Befehls per Hand zu kontrollieren. D. h. `\mkccitation` wird genutzt, um bei Bedarf einen Abstand hinzuzufügen, den Rest aber bei dem Literaturstellenbefehl zu verlassen. Im Falle von `biblatex` läuft das auf Folgendes hinaus:

```
\SetCiteCommand{\parencite}
\renewcommand{\mkccitation}[1]{ #1}
```

für Klammerliteraturstellen und

```
\SetCiteCommand{\footcite}
\renewcommand{\mkccitation}[1]{#1}
```

für Literaturstellen in Fußnoten.

## 10 Hinweise und Vorsorge

### 10.1 Eingabecodierungen

Das durch dieses Paket bereitgestellte aktive Zitieren kann unter bestimmten Umständen von den `inputenc`/`inputenx`-Paketen abhängig sein oder unterstützt werden. Solange die aktiven Zitate im Bereich von 0 bis 127 liegen, gibt es keinen Vorteil beim Laden von `inputenc`. Wenn sie einen 8-Bit-Eingabecode wie `latin1` benutzen, ist `inputenc` für das Funktionieren von Zitaten im geeigneten *verbatim*-Kontext erforderlich. Es sollte daher geladen werden, bevor alle aktiven Zitate zugeordnet wurden (nicht notwendigerweise vor dem Laden von `csquotes`). Die Macro-Ebene UTF-8-Unterstützung dieses Pakets baut auf dem `utf8`-Modul des `inputenc`-Pakets auf. Wenn diese Codierung genutzt wird, müssen Sie sicherstellen, dass `inputenc` mit der `utf8`-Option geladen ist. Verwenden Sie nicht die `utf8x`-Option, da diese implizit das `ucs`-Paket lädt, welches nicht von `csquotes` unterstützt wird. Die UTF-8-Codierung wird automatisch erkannt. Alle Befehle, welche in § 4 diskutiert wurden, arbeiten für gewöhnlich mit dieser Codierung. XeTeX und LuaTeX im nativen UTF-8-Modus arbeiten ebenfalls wie erwartet und erfordern keine zusätzlichen Pakete. Siehe auch § 10.5.

### 10.2 Ausgabe-Codierungen

Die OT1-Schriftartcodierung, welche die Standardcodierung von LaTeX ist, enthält lediglich die Anführungszeichen, welche im Englischen genutzt werden. Sie benöti-

Doppelte Anführungszeichen		Einfache Anführungszeichen	
Befehl	Beispiel	Befehl	Beispiel
<code>\textquotedblleft</code>	“AaGg“	<code>\textquoteleft</code>	‘AaGg’
<code>\textquotedblright</code>	”AaGg”	<code>\textquoteright</code>	’AaGg’
<code>\quotedblbase</code>	„AaGg„	<code>\quotesinglbase</code>	,AaGg,
<code>\guillemotleft</code>	«AaGg«	<code>\guilsinglleft</code>	‹AaGg‹
<code>\guillemotright</code>	»AaGg»	<code>\guilsinglright</code>	›AaGg›

Tabelle 6: Anführungszeichen, die von den Codierungen T1, LY1, EU1/EU2 bereitgestellt werden

gen ein Codierung wie T1, LY1, oder mit XeTeX oder LuaTeX EU1/EU2, um Rechtsschreibvorschläge oder grundlegende Anführungszeichen zu erhalten. Dieses Paket hält bewusst die Bereitstellung von Problemumgehungen für die OT1-Altcodierung zurück. Falls Sie T1 oder andere erweiterte Codierungen für einige der Anführungszeichen benötigen, werden Sie es höchstwahrscheinlich brauchen, um die ordnungsgemäße Silbentrennung für die jeweilige Sprache zu erhalten. In Tabelle 6 können Sie eine Liste der häufigsten Anführungszeichen eingeschlossen in T1, LY1, und EU1/EU2 sehen.

### 10.3 Gültige aktive Zitate

Im Allgemeinen kann ein aktives Zitat jedes einzelne Zeichen mit dem Kategoriecode 12 oder 13 oder ein von einer Multibyte UTF-8-Sequenz repräsentiertes einzelnes Zeichen sein. Es gibt jedoch ein paar Ausnahmen: Zahlen, Satzzeichen, der Apostroph, der Bindestrich und alle Zeichen, die Teil der LaTeX-Syntax sind, werden zurückgewiesen. Letztendlich werden die folgenden Zeichen berücksichtigt, indem sie in diesem Paket belegt wurden: A-Z a-z 0-9 . , ; : ! ? ' - # \$ % & ^ \_ ` ~ \ @ \* { } [ ]

### 10.4 Ungültiges Schachteln und unsymmetrische aktive Zitate

Jedes Zitat bildet eine Gruppe, die sowohl die zitierte Textstelle als auch die Anführungszeichen enthält. Dieses Paket bildet die Schachtelungsebene aller Zitate und sorgt so für grundlegende Gültigkeit. Wenn Zitate in einer ungültigen Weise verschachtelt sind, wird eine Fehlermeldung ausgegeben. Denken Sie daran, dass die aktiven Zitate mehr als ein bequemer Weg sind, um Anführungszeichen einzugeben. Sie sind eigenständige Markup-Elemente, welche ebenso Gruppierungen implizieren. Daher müssen sie ausgeglichen sein und dürfen sich nicht mit den Grenzen anderer Gruppen vermischen. Dieses Paket sorgt dafür, dass ein Fehler ausgelöst wird, wenn die Zitate unausgewogen oder auf eine ungültige Weise verschachtelt

sind. Beachten Sie allerdings, dass Pakete in der Regel weder Fehler niederen Levels, die durch Gruppierungsfehler aufgetreten sind, abgefangen werden können, noch Kontrolle über die Formulierung allgemeiner Fehlermeldungen haben.

## 10.5 Aktive Zitate in speziellen Zusammenhängen

Alle Zitatbefehle sind für den Einsatz im Textmodus konzipiert und werden eine Fehlermeldung im mathematischen Modus ausgeben. Beachten Sie, dass alle aktiven Zitate ihre ursprüngliche Funktion im mathematischen Modus beibehalten. Es ist durchaus möglich, ein Zeichen wie das Größer-als-Symbol als aktives Zitat ohne Störung im mathematischen Modus zu benutzen. Im wörtlichen Zusammenhang werden die aktiven Zitate normalerweise deaktiviert. Wenn sich ein Zeichen im Bereich 128–255 befindet, wird seine ursprüngliche Funktion wieder hergestellt, sodass das `inputenc/inputenx`-Paket es in der wörtlichen Umgebung behandeln kann. Diese Funktion wird mit der wörtlichen Standardumgebung bereitgestellt und mit dieser von den Paketen `verbatim`, `fancyvrb`, `moreverb`, und `alltt` zur Verfügung gestellt (oder definiert). Dies gilt auch für den `\verb`-Befehl und das `shortvrb`-Paket. Das `listings`-Paket stellt bestimmte Unterstützungen für erweiterte Eingabecodes zur Verfügung. Bei Verwendung dieses Pakets aktivieren Sie die „erweiterte Zeichen“-Option und legen die Eingabecodierung fest. Beachten Sie, dass `listings` nicht das Macro-Level UTF-8-Entschlüsselung unterstützt. XeTeX und LuaTeX im nativen UTF-8-Modus arbeiten wie erwartet und erfordern nicht das `inputenc`-Paket. Einige Vorsicht ist weiterhin bei der Wahl aktiver Zitate erforderlich. Beachten Sie, dass Sie normalerweise aktive Zeichen nicht als Argument für Befehle erwarteter Zeichenreihen benutzen können, wie zum Beispiel `\input`, `\label`, oder `\cite`. Es gibt zwei Pakete, welche versuchen, dieses Problem zu beheben: das `babel`-Paket und das `underscore`-Paket (wenn es mit der `strings`-Option geladen wurde). Beide Pakete definieren mehrere Standardbefehle neu, die von diesem allgemeinen Problem betroffen sind. Wenn eines dieser Pakete geladen ist, wird `csquotes` alle Vorteile dieser Verbesserungen automatisch benutzen. Leider decken beide Pakete einen unterschiedlichen Satz von Befehlen ab und keines alle möglichen gefährdeten Befehle.

## 10.6 PDF-Strings und `hyperref`-Unterstützung

Diese Paketschnittstellen mit dem `hyperref`-Paket als PDF erzeugt Strings als Lesezeichen. Siehe § 8.5, um Anführungszeichen zu konfigurieren und in PDF-Strings zu verwenden. Unterstützung für PDF-Strings kann nur mit der Grundausstattung bereitgestellt werden, wie in § 3.1, § 3.2, § 4.1 und § 4.2 zu sehen. Beachten Sie, dass die Art und Weise wie `hyperref` PDF-Strings baut, schwere Beschränkungen auf die Fähigkeiten aller Befehle verhängt. Vor allem kann die Schachtelungsebene von Zitaten in diesem Zusammenhang nicht verfolgt werden. Geschachtelte Zitate erhalten in der Regel äußere Marken, aber Sie können mit einem Sternzeichen

versehene Befehle oder aktive innere Zitate verwenden, um innere Marken explizit anzufordern. Falls es Anführungszeichen gibt, die in den Dokumenteigenschaften einer PDF-Datei enthalten sind, müssen Sie `\hypersetup` verwenden, um einen String zu spezifizieren. Der Ersatzmechanismus wird nicht innerhalb des optionalen Arguments von `\usepackage` funktionieren. Siehe im `hyperref`-Handbuch für weitere Details.

## 10.7 Fußnoten innerhalb von Zitaten

Dieses Paket setzt automatische die Schachtelungsebene innerhalb einer Fußnote, die in einem Zitat enthalten ist, zurück. Wenn das `babel`- oder `polyglossia`-Paket geladen wurde, wird es auch die Spracheinstellungen zurücksetzen. Die Sprache des Fußnotentextes einschließlich des Silbentrennungsschemas passt zum umgebenden Text des Zitates. Dies gilt für `Parboxen`, `Minipages` sowie `Floats`.

## 10.8 Benutzung von `csquotes` mit der `babel`-Kurzschrift

Die in § 11.5 besprochenen Befehle können mit Kürzeln aus dem Paket `babel` kombiniert werden, so dass `babel` die Benutzeroberfläche unterstützt und `csquotes` das Back-End (nachgeschaltet).<sup>4</sup> Zum Beispiel definiert das deutsche Module vom `babel`-Paket unter anderem die Kürzel `"`` and `"'`. Solche Kürzel sind Eingabehilfen, das heißt physikalische Markup-Elemente mit einer festen Definition. Das geschriebene `"`` ist kurz gesagt `\quotedblbase`, aber im Konzept nichts anderes. Diese Kürzel können in ‚smart quotes‘ umgewandelt werden, welche sich wie `\enquote` verhalten. Hier ist eine einfache Ad-hoc-Lösung, die für Dokumente mit nur einer Sprache geeignet ist:

```
\documentclass{...}
\usepackage[german]{babel}
\usepackage[babel=once]{csquotes}
\defineshorthand{"`}{\openautoquote}
\defineshorthand{"'}{\closeautoquote}
```

Es ist möglich, solche Definitionen in `csquotes.cfg` einzubeziehen. In diesem Fall ist der Code etwas komplexer, da mehr Allgemeinheit benötigt wird:

```
\AtEndPreamble{%
  \@ifpackageloaded{babel}
  {\iflanguage{german}
    {\declare@shorthand{german}{"`}{\openautoquote}%
    \declare@shorthand{german}{"'}{\closeautoquote}}}
```

<sup>4</sup> Dies trifft auch auf das `polyglossia`-Paket zu, wenn die `babelshorthands`-Option von `polyglossia` aktiviert wurde.

```
{}
```

Dieser Code definiert Kürzel nur, wenn das `babel`-Paket geladen wurde und die Hauptsprache `german` ist. Beachten Sie, dass `babel`-Kürzel sprachspezifisch sind. Die Art, wie sie konfiguriert und gehandhabt werden, ist technisch und konzeptionell verschieden von dem aktiven Zitieren, wie es in § 4 diskutiert wurde. Aktive Zitate werden global definiert und werden automatisch an die aktuelle Sprache angepasst. Mit `babel` hat jede Sprache einen eigenen Satz von Kürzeln. Beachten Sie auch, dass `babel \AtBeginDocument` verwendet, um die Hauptdokumentensprache zu initialisieren, einschließlich sinngemäßen Kürzeln. Wir verwenden `\AtEndPreamble`, um den Code ans Ende der Präambel zu verschieben. Auf diese Weise können wir sicher sein, dass `babel` geladen wurde, aber die Hauptdokumentensprache noch nicht initialisiert wurde. Siehe das `babel`-Handbuch für weitere Details. Der `\AtEndPreamble`-Befehl wird vom `etoolbox`-Paket bereitgestellt.

## 10.9 Verschiedene Hinweise für vordefinierte Stile

Es gibt drei Arten, die einem besonderen Zweck dienen: `default`, `fallback` und `debug`. Der `default`-Stil ist ein dynamisches Pseudonym, das Standardzitatstile verwendet, wenn die mehrsprachige Schnittstelle nicht aktiviert wurde. Die Paketoption `style` und der Befehl `\setquotestyle` definieren dieses Pseudonym neu. Der `fallback`-Stil ist ein Back-End-Stil, der als Sicherung genutzt wird, wenn die mehrsprachige Schnittstelle aktiviert ist, aber kein passender Zitatstil für die gegenwärtige Sprache vorliegt.

Durch die Voreinstellung werden Fragezeichen fett gedruckt. Der `debug`-Stil druckt keine Anführungszeichen, aber das aktuelle `Zitatlevel` wird als fettgedruckte Nummer ausgegeben. Dieser Stil kann mit der `style`-Option oder mit dem `\setquotestyle`-Befehl ausgewählt werden. Es ist für die Fehlersuche gedacht. Alle Varianten des `french`-Stils verwenden ausschließlich Guillemets als äußere Zeichen. Die Stilvariante `quotes` benutzt doppelte Anführungszeichen als innere Zeichen. Die mit einem Sternzeichen versehene Variante `quotes*` ist ähnlich seinem regulären Gegenstück, außer dass es auch ausschließlich ein inneres Zeichen ist. Die `guillemets`-Variante beschäftigt sich auf allen Ebenen mit Guillemets. Auch fügt sie Guillemets am Anfang eines jeden Paragraphen innerhalb eines Zitats, welches mehrere Paragraphen umfasst, ein. Darüber hinaus werden zwei benachbarte Zeichen am Ende eines Zitats durch ein einziges ersetzt; wenn zwei verschachtelte Zitate gleichzeitig enden, wird das zweite Schlusszeichen automatisch weggelassen. Die mit einem Sternzeichen versehene Variante `guillemets*` ist ähnlich seinem regulären Gegenstück, die sich nur im mittleren Zeichen, welches zu Beginn jedes Paragraphen eingefügt wurde, unterscheiden. Die reguläre Variante benutzt einen nach links weisenden Guillemet, wobei das mit dem Stern versehene einen nach rechts weisenden benutzt.

## 11 Autorenschnittstelle

Die folgenden Abschnitte behandeln die Programmierschnittstellen im `csquotes`-Paket sowie einige Details zur Umsetzung. Sie sind Klassen- und Paketautoren zuge-  
dacht, welche sich diesem Paket anschließen möchten.

### 11.1 Aktive Zitate kontrollieren

Die Autorbefehle in diesem Kapitel verhalten sich im Wesentlichen wie die entsprechenden Benutzerbefehle, die in § 4.5 diskutiert wurden. Der einzige Unterschied besteht darin, dass diese ruhig im Hintergrund arbeiten, ohne das Schreiben von Mitteilungen an die Kopiedateien. Der Gültigkeitsbereich dieser Befehle ist lokal, sodass alle Änderungen auf eine Gruppe beschränkt werden können. Achten Sie darauf, dass das aktive Zitieren zu Beginn des Dokumentenkörpers aktiviert ist. Unter keinen Umständen wird dieses Paket Zeichen in der Dokument-Präambel aktivieren. Sie benötigen nur die folgenden Befehle im Umgang mit dem aktiven Zitieren zu Beginn oder im Dokumentenkörper.

`\@enablequotes` Dieser Befehl ermöglicht es, allen Zeichen aktive Zitate zuzuordnen. Darüber hinaus werden deren Definitionen wiederhergestellt, falls sie funktionsunfähig oder versehentlich überschrieben worden sind.

Mit einer Single-Byte-Codierung definiert dieser Befehl alle zugewiesenen Zeichen (neu) und aktiviert sie. Mit der UTF-8-Codierung wird das interne Makro `neu` definiert, welches vom `inputenc`-Paket verwendet wird, um die jeweilige UTF-8-Sequenz (`\u8:<character>`) zu setzen. UTF-8 Zeichen im Bereich 0–127 werden als Single-Byte-Codierungen behandelt. Bei Verwendung einer TeX-Maschine mit nativer UTF-8-Unterstützung, wie zum Beispiel XeTeX, werden alle Zeichen wie mit einer Single-Byte-Codierung behandelt.

`\@disablequotes` Dieser Befehl stellt den *status quo ante* aller aktiven Zitate wieder her. Mit der Single-Byte-Codierung gibt es zwei Fälle. (1) Wenn ein Zeichen bereits aktiviert wurde, als ihm ein aktives Zitat zugeordnet wurde, wird dessen frühere Definition wiederhergestellt. (2) Falls ein Zeichen noch nicht aktiviert wurde, als es zugeordnet wurde, wird sein ehemaliger Kategoriecode wiederhergestellt. Mit der UTF-8-Codierung stellt der Befehl die frühere Definition des internen Makros wieder her, welches vom `inputenc`-Paket verwendet wird, um die jeweilige UTF-8-Sequenz zu setzen. UTF-8-Zeichen im Bereich 0–127 werden wie mit einer Single-Byte-Codierung behandelt. Bei Verwendung einer TeX-Maschine mit nativer UTF-8-Unterstützung, wie zum Beispiel XeTeX, werden alle Zeichen wie mit einer Single-Byte-Codierung behandelt.

`\@verbatimquotes` Für wörtliche Umgebungen und ähnliche Anwendungen verwenden Sie diesen Befehl anstatt `\@disablequotes`. Es definiert die aktiven Zitate auf eine Weise, die besser für die wörtliche Zeichensetzung geeignet ist. Mit der Single-Byte-Codierung

wird eines der folgenden Dinge getan. (1) Falls ein Zeichen sich im Bereich 0–127 befindet, wird es so neu definiert, dass es sich selbst dem Kategoriecode 12 erweitert. (2) Wenn sich ein Zeichen im Bereich 128–255 befindet, gibt es zwei Möglichkeiten. (a) Wenn ein Zeichen bereits aktiviert ist, als ihm ein aktives Zitat zugeordnet wurde, wird dessen frühere Definition wiederhergestellt. (b) Falls ein Zeichen nicht zuvor aktiviert wurde, wird es so neu definiert, dass es sich selbst mit seinem früheren Kategoriecode erweitert.

Zeichen im Bereich 0–127 wurden der `\dospecials`-Liste hinzugefügt. Zeichen im Bereich 128–255 bleiben aktiv und erlauben dem `inputenc`-Paket, sie wörtlich zu setzen (aufgrund von Fall 2a, welcher besagt, dass Sie `inputenc` laden müssen, bevor die aktiven Zitate zugeordnet werden). Fall 2b ist gewöhnlich in der wörtlichen Umgebung nicht wünschenswert. Doch wenn `inputenc` geladen wurde, sollte dies nicht passieren. Mit der UTF-8-Codierung definiert der Befehl die frühere Definition des internen Makros neu, welches vom `inputenc`-Paket verwendet wird, um die jeweilige UTF-8-Sequenz zu setzen. UTF-8-Zeichen im Bereich 0–127 werden wie mit einer Single-Byte-Codierung behandelt. Bei Verwendung einer TeX-Maschine mit nativer UTF-8-Unterstützung, wie zum Beispiel XeTeX, werden alle Zeichen wie mit einer Single-Byte-Codierung behandelt.

Aufgrund von Fall 1 ist `\@verbatimquotes` unabhängig von jedem `\dospecials`-Prozess. Sie können alle aktiven Zitate ausschließlich mit diesem Befehl wörtlich setzen. Der Vorteil dieses Ansatzes ist, dass es keiner Kategoriecodeänderung bedarf, damit der Befehl auch verwendet werden kann, um ein Argument zu ändern, nachdem es gelesen wurde. Beachten Sie auch, dass die wörtliche LaTeX-Standard-Umgebung sowie alle Umgebungen, die zur Verfügung gestellt oder über die Pakete `verbatim`, `fancyvrb`, `moreverb`, und `alltt` definiert sind, automatisch geliefert werden. Das trifft auf den `\verb`-Befehl und das `shortvrb`-Paket zu.

`\@deletequotes` Die Befehl führt implizit `\@disablequotes` aus und gibt alle aktiven Zitate frei, die aus einem kompletten Zurücksetzen aller aktiven Zitate resultieren, sodass sie neu definiert werden. Dieser Befehl sollte mit Vorsicht verwendet werden, da das Zurücksetzen für den Benutzer nicht sichtbar ist. Die Benutzung von `\DeleteQuotes` kann von Vorteil sein.

## 11.2 Aktive Zitate in einem aus Strings bestehenden Kontext

Ein mögliches Problem mit aktiven Zitaten können Kontexte sein, die nur aus Strings bestehen. Das sind Fälle, in welchen ein aktives Zeichen zur Bildung eines Kontrollsequenznamens verwendet wird. Ein typisches Beispiel ist der `\label`-Befehl, welcher ein String von Zeichen erwartet. Jedes aktive Zeichen kann `\label` abbrechen, wenn es in dessen Argument verwendet wird. Es gibt zwei Pakete, die versuchen, diese Situation in Ordnung zu bringen, wenn auch auf verschiedenen Wegen: `babel` und `underscore`.

Das babel-Paket definiert den Schalter `\if@safe@actives` und setzt einige Standardbefehle so ein, dass der Schalter auf `true` gestellt wird, während sie ihre Argumente prozessieren. Der erste Schritt, der vom `underscore`-Paket vorgenommen wird, ist etwas anders. Wenn `underscore` mit der `strings`-Option geladen wurde, werden während der Prozessierung der Argumente einige Befehle eingesetzt, sodass `\protect` zu `\string` äquivalent ist. Wenn eines dieser Pakete geladen ist, wird `csquotes` automatisch die Vorteile davon nutzen. Leider setzen beide Pakete eine unterschiedliche Menge von Befehlen ein und keines von ihnen umfasst alle möglichen angreifbare Befehle. Wenn `babel` geladen wurde, können Sie beispielsweise aktive Zitate in einem Argument von `\label` verwenden, allerdings nicht in einem Argument von `\input`. Wenn Sie `underscore` mit seiner Option `strings` geladen haben, können aktive Zitate auch im Argument von `\input` genutzt werden.

Beim Schreiben eines Pakets, welches vom Benutzer angegebene Argumente in einem nur aus Strings bestehenden Kontext prozessieren muss, gibt es zwei Möglichkeiten, um sich mit aktiven Zitaten zu befassen. Unter dem Ansatz von Paket `babel` können Sie Folgendes machen:

```
\let\if@safe@actives\iftrue
```

Dies wird am besten in einer Gruppe durchgeführt. Wenn die Gruppierung nicht möglich ist, müssen Sie sicherstellen, dass der Schalter ordnungsgemäß wiederhergestellt wurde. Im Gegensatz zur Benutzung von `\@safe@activestrue` funktioniert dieser Ansatz sogar, wenn `babel` nicht geladen ist. Doch beachten Sie, dass in diesem Fall drei Zustände bei der Wiederherstellung des Schalter berücksichtigt werden müssen: wahr, falsch und undefiniert. Wenn der Ansatz vom `underscore`-Paket verwendet wird, können Sie Folgendes tun:

```
\let\@protect\protect \let\protect\string
```

Dies könnte entweder in einer Gruppe oder ohne Gruppierung gemacht werden, wobei aber `\restore@protect` folgt. Der erste Ansatz arbeitet mit aktiven Zeichen des `babel`- und des `underscore`-Pakets. Das zweite arbeitet mit den Paketen `underscore` und `at`. Leider unterstützen die aktiven Zeichen des `inputenc`-Pakets keine der oben genannten Techniken. Soweit es `csquotes` betrifft, spielt es keine Rolle, welchen Ansatz Sie nehmen. In beiden Fällen werden alle aktiven Zitate mit Kategoriecode 12 erweitert. Die Marco-Ebene unterstützt UTF-8, verschlüsselt UTF-8 aktive Zitate, die zu einem String von Zeichen mit Kategoriecode 12 erweitert werden. Dieser String wird UTF-8 gültig werden. In einer wörtlichen `\write`-Operation sollten Sie lieber eine der in diesem Abschnitt behandelten Techniken anwenden als `\@verbatimquotes`, welche für die wörtliche Schriftsetzung gedacht ist.

### 11.3 Blockzitate

Die Blockzitatmöglichkeiten benötigen zum Setzen alle Zitate zweimal. Der erste Durchlauf ist erforderlich, um die Länge des Zitats zu messen. Das eigentliche Setzen findet im zweiten Durchlauf statt und ist formatabhängig vom Ergebnis des ersten. Um die Nebenwirkungen des ersten (Test-)Durchlaufs zu verhindern, macht das `csquotes`-Paket folgendes: (1) Durchführung des ersten Durchlaufs in einer Gruppe, (2) Anlegen von Kontrollpunkten, um alle  $\TeX$ -Zähler einzufrieren, und (3) Setzen von `\if@files` auf `false`. Es kann aber keine Nebenwirkungen verhindern, die durch Befehle hervorgerufen werden, welche (1) globale Zuweisungen vornehmen, die im zweiten Durchlauf nicht überschrieben werden können (z. B. durch `\g@addto@macro`), (2) Zähler auf eine Weise global erhöhen, sodass LaTeX-Zählerbefehle umgangen werden, oder (3) `\if@files` nicht jedes Mal prüfen, wenn sie gerade dabei sind, eine Hilfsdatei zu schreiben. Wenn Sie eine Störungen im Zusammenhang mit dem Testdurchlauf beobachten (z. B. wenn Zähler zweimal erhöht werden oder ein Element zweimal in einer Liste erscheint), verwenden Sie `\BlockquoteDisable`, um den betroffenen Befehl neu zu definieren oder ihn vorübergehend auszuschalten .

```
\BlockquoteDisable{<code>}
```

Der `<code>` kann ein beliebiger LaTeX-Code sein, der anfällige Befehle lokal neu definiert, sodass sie während des Testdurchlaufs anders arbeiten. Der `<code>` sollte natürlich selbst keine globalen Aufgaben beinhalten. Diese Lösung sollte als letztes Ressort betrachtet werden, aber kann der schnellste Weg sein, um ein anfälliges Paket festzulegen. Beachten Sie, dass es keine Notwendigkeit gibt, Parameterzeichen durch ihre Verdopplung im `<code>`-Argument zu entziehen. Verwenden Sie einfach diesen Befehl wie `\AtBeginDocument` und Ähnliches.

### 11.4 Registrierung von Anführungszeichen

Um Satzzeichen innerhalb von Zitaten zu verfolgen, fordert dieses Paket, dass alle Anführungszeichen transparent für den Raumfaktor sind, d. h. sie müssen einen Raumfaktorcode von Null haben. Diese Einstellung ist spezifisch für die Codierung der Ausgabe. Einstellungen für die Codierungen OT1, OT2, OT4, T1, LY1, LGR, T2A, T2B, T2C, LCY, X2, und EU1/EU2 sind standardmäßig vorgesehen. Andere Codierungen können durch folgenden Befehl in der Konfigurationsdatei festgelegt werden:

```
\DeclareQuoteGlyph{<encoding>}{<position>}
```

Der `<encoding>` ist der Name der Ausgabecodierung, z. B. OT1. Dieser String entspricht den Bezeichnern, die genutzt werden, wenn das `fontenc`-Paket geladen

wird. Das  $\langle position \rangle$ -Argument ist ein Integer, der die Position eines Symbols in dieser Codierung kennzeichnet. Sie müssen alle Anführungszeichen in der Ausgabecodierung registrieren, indem Sie eine Deklaration für jedes Symbol verwenden. Als Beispiel sind dies die Einstellungen für die OT1-Codierung:

```
\DeclareQuoteGlyph{OT1}{34}% = \textquotedblright  
\DeclareQuoteGlyph{OT1}{39}% = \textquoteright  
\DeclareQuoteGlyph{OT1}{92}% = \textquotedblleft  
\DeclareQuoteGlyph{OT1}{96}% = \textquoteleft
```

Das  $\langle position \rangle$ -Argument kann in jeder Schreibweise von TeX in Integeraufgaben akzeptiert werden, z. B. 171 in dezimaler oder "00AB in hexadezimaler Schreibweise. In den Einstellung in `csquotes.def` können Sie weitere Beispiele nachlesen. Der Vorteil der Zitatregistrierung mit dem oben genannten Befehl (anstatt der Anpassung des globalen Raumfaktorcodes) ist, dass die Deklarationen nur lokal innerhalb der Zitate verwendet werden und keine Auswirkungen auf andere Teile des Dokuments haben.

## 11.5 Automatische Anführungszeichen

Die Befehle in diesem Abschnitt stellen den Zugang zu den automatischen Zitatemöglichkeiten auf einem etwas niedrigeren Niveau als die Benutzerbefehle in § 3.1 und § 4.1 bereit. Im Gegensatz zu den in § 11.6 behandelten Befehlen sind die Möglichkeiten in diesem Abschnitt vollwertige Markup-Elemente, die die Schachtelungsebene überprüfen und einen Fehler ausgeben, wenn Zitate auf ungültige Weise geschachtelt sind. Sie bilden Gruppen und müssen immer ausbalanciert sein. Siehe § 10.4 für weitere Details. Mit anderen Worten, die Möglichkeiten in diesem Abschnitt sind semantische Markup-Elemente, die in § 11.6 sind physikalische Markup-Elemente.

`\openautoquote` Öffnen eines ineinander verschachtelten Zitats.  
`\closeautoquote` Schließen eines ineinander verschachtelten Zitats.

In Bezug auf ihre Funktion entsprechen die oben genannten Befehle den regulären Versionen von `\enquote` und `\MakeAutoQuote`. Die folgenden Befehle entsprechen den mit einem Sternzeichen versehenen Varianten `\enquote*` und `\MakeAutoQuote*`:

`\openinnerquote` Öffnen eines inneren Zitats.  
`\closeinnerquote` Schließen eines inneren Zitats.

Die oben genannten Befehle können verwendet werden, um eine alternative Benutzeroberfläche zu implementieren. Beispielsweise können Sie sie mit den Kürzeln des `babel`-Pakets kombinieren, sodass `babel` die Benutzeroberfläche und `csquotes` das Back-End bereitstellt. Siehe § 10.8 und das `babel`-Handbuch für Details.

## 11.6 Interne Anführungszeichen

Die Befehle in diesem Abschnitt drucken die Anführungszeichen des aktuellen Stils, wie definiert mit `\DeclareQuoteStyle`, ohne jegliche Gruppierung oder Verschachtelungskontrolle. Die Anführungszeichen betrachten alle Veränderung des Zitatstils. Wenn die mehrsprachige Schnittstelle aktiviert ist, werden sie auch mit der aktuellen Sprache synchronisiert.

- `\textooquote` Druckt die Öffnung äußerer Anführungszeichen des gerade aktiven Zitatstils.
- `\textcoquote` Druckt die Schließung äußerer Anführungszeichen.
- `\textmoquote` Druckt die Mitte äußerer Anführungszeichen.
- `\textoiquote` Druckt die Öffnung äußerer Anführungszeichen.
- `\textciquote` Druckt die Schließung innerer Anführungszeichen.
- `\textmiquote` Druckt die Mitte innerer Anführungszeichen.

Beachten Sie, dass die Initialisierungshaken für den jeweiligen Zitatstil nicht automatisch ausgeführt werden. Sie können separat aufgerufen werden:

- `\initoquote` Ausführen der äußeren Initialisierungshaken.
- `\initiquote` Ausführen der inneren Initialisierungshaken.

Der Umfang dieser Haken muss immer auf eine Gruppe beschränkt werden.

## 12 Geschichte der Revision

Diese Revisionsgeschichte ist eine Liste von Änderungen, die relevant für die Nutzer dieses Pakets sind. Änderungen eher technischer Natur, welche nicht die Benutzeroberfläche oder die Handlungsweise des Pakets beeinflussen, sind nicht in der Liste enthalten. Wenn ein Eintrag in der Revisionsgeschichte besagt, dass eine Funktion *extended* wurde, deutet dies auf eine syntaktisch abwärtskompatible Modifikation hin, wie das Hinzufügen eines optionalen Arguments zu einem bestehenden Befehl. Einträge geben an, dass eine Funktion *modified*, *renamed* oder *removed* wurde, sie erfordern Aufmerksamkeit. Sie weisen auf eine Modifikation hin, die Änderungen an bestehenden Dokumenten in einigen, hoffentlich seltenen Fällen erforderlich machen können. Die `version`-Option von § 2.1 kann in diesem Fall hilfreich sein. Die Zahlen auf der rechten Seite zeigen den entsprechenden Abschnitt des Handbuchs.

### 5.1 2010-11-19

Hinzugefügte Paketoption `debug` . . . . . 2.1

Hinzugefügte Paketooption <code>threshold</code> . . . . .	2.1
Hinzugefügte Paketooption <code>thresholdtype</code> . . . . .	2.1
Hinzugefügte Paketooption <code>parthreshold</code> . . . . .	2.1
Hinzugefügte Paketooption <code>splitcomp</code> . . . . .	2.1
Verbesserte Blockzitatmöglichkeiten . . . . .	3.5
Verbesserte integrierte Blockzitatmöglichkeiten . . . . .	5.3
<b>5.0c 2010-09-21</b>	
Paketooption <code>csdisplay</code> hinzufügen . . . . .	2.1
<code>\csdisplaytrue</code> und <code>\csdisplayfalse</code> hinzufügen . . . . .	2.1
Vorgegebene Konflikte mit <code>polyglossia</code>	
<b>5.0b 2010-08-06</b>	
Vorgegebene Probleme mit <code>babel</code> 's aktiven Satzzeichen	
<b>5.0a 2010-06-09</b>	
Vorgegebene Fehler in Zusammenhang mit, in der Mitte stehenden, Anführungszeichen	
Kleinere interne Updates für <code>biblatex</code>	
<b>5.0 2010-06-02</b>	
Umbenannte Paketooption von <code>babel</code> zu <code>autostyle</code> . . . . .	2.1
Hinzugefügte Kompatibilitätsoption <code>babel</code> . . . . .	2.2
Hinzugefügte vorbereitende <code>polyglossia</code> -Schnittstelle . . . . .	2.1
Hinzugefügte Paketooption <code>autopunct</code> . . . . .	2.1
Hinzugefügte Paketooption <code>version</code> . . . . .	2.2
Hinzugefügtes <code>\hybridblockquote</code> . . . . .	3.6
Hinzugefügtes <code>\hybridblockcquote</code> . . . . .	5.4
Hinzugefügtes <code>\MakeHybridBlockQuote</code> . . . . .	4.4
Hinzugefügtes <code>\DeclareAutoPunct</code> . . . . .	8.9
Erweitertes <code>\textquote</code> . . . . .	3.3
Erweitertes <code>\foreigntextquote</code> . . . . .	3.4
Erweitertes <code>\hyphentextquote</code> . . . . .	3.4
Erweitertes <code>\blockquote</code> . . . . .	3.5
Erweitertes <code>\foreignblockquote</code> . . . . .	3.6

Erweitertes <code>\hyphenblockquote</code> . . . . .	3.6
Erweitertes <code>\textcquote</code> . . . . .	5.1
Erweitertes <code>\foreigntextcquote</code> . . . . .	5.2
Erweitertes <code>\hyphentextcquote</code> . . . . .	5.2
Erweitertes <code>\blockcquote</code> . . . . .	5.3
Erweitertes <code>\foreignblockcquote</code> . . . . .	5.4
Erweitertes <code>\hyphenblockcquote</code> . . . . .	5.4
Hinzugefügtes <code>\mkttextquote</code> . . . . .	8.7
Hinzugefügtes <code>\mkblockquote</code> . . . . .	8.7
Hinzugefügtes <code>\mkbegdispquote</code> . . . . .	8.7
Hinzugefügtes <code>\mkenddispquote</code> . . . . .	8.7
Entferntes <code>\mkpretextpunct</code> . . . . .	8.7
Entferntes <code>\mkmidtextpunct</code> . . . . .	8.7
Entferntes <code>\mkfintextpunct</code> . . . . .	8.7
Entferntes <code>\mkpreblockpunct</code> . . . . .	8.7
Entferntes <code>\mkmidblockpunct</code> . . . . .	8.7
Entferntes <code>\mkfinblockpunct</code> . . . . .	8.7
Entferntes <code>\mkpredisppunct</code> . . . . .	8.7
Entferntes <code>\mkmiddisppunct</code> . . . . .	8.7
Entferntes <code>\mkfindisppunct</code> . . . . .	8.7
Entferntes <code>\quotetext</code> . . . . .	8.7
Entferntes <code>\quoteblock</code> . . . . .	8.7
<code>\ifquotepunct</code> umbenannt zu <code>\ifpunct</code> . . . . .	8.8
<code>\ifquoteterm</code> umbenannt zu <code>\ifterm</code> . . . . .	8.8
Hinzugefügtes <code>\ifpunctmark</code> . . . . .	8.8
Entferntes <code>\ifquotecolon</code> . . . . .	8.8
Entferntes <code>\ifquotecomma</code> . . . . .	8.8
Entferntes <code>\ifquoteexclam</code> . . . . .	8.8
Entferntes <code>\ifquoteperiod</code> . . . . .	8.8
Entferntes <code>\ifquotequestion</code> . . . . .	8.8
Entferntes <code>\ifquoteseMIColon</code> . . . . .	8.8
<code>\ifstringblank</code> umbenannt zu <code>\ifblank</code> . . . . .	8.8

Hinzugefügtes <code>\unspace</code> . . . . .	8.8
Hinzugefügtes <code>\textelp</code> . . . . .	7
Hinzugefügtes <code>\textins</code> . . . . .	7
Hinzugefügtes <code>\mktextelp</code> . . . . .	8.10
Hinzugefügtes <code>\mktextelpins</code> . . . . .	8.10
Hinzugefügtes <code>\mktextinselp</code> . . . . .	8.10
Hinzugefügtes <code>\mktextins</code> . . . . .	8.10
Hinzugefügtes <code>\mktextmod</code> . . . . .	8.10
Erweiterte Dokumentation . . . . .	9.2
Entferntes <code>\cquote</code> -Vermächtnispseudonym (stattdessen <code>\textcquote</code> benutzen) 5.1	
Entferntes <code>\foreigncquote</code> -Vermächtnispseudonym (use <code>\foreigntextcquote</code> ) 5.2	
Entferntes <code>\hyphencquote</code> -Vermächtnispseudonym (use <code>\hyphentextcquote</code> ) 5.2	
Entferntes <code>\RestoreQuotes</code> -Vermächtnispseudonym (use <code>\EnableQuotes</code> )	4.5
Entferntes <code>\@restorequotes</code> -Vermächtnispseudonym (use <code>\@enablequotes</code> ) 11.1	
Hinzugefügte Unterstützung für die EU2-Codierung . . . . .	11.4
Hinzugefügter Zitatstil für Kroatisch <b>4.4d 2010-02-06</b>	
Hinzugefügter Zitatstil für Niederländisch	
Hinzugefügter Zitatstil für Finnisch	
Hinzugefügter Zitatstil für Griechisch	
Hinzugefügte Unterstützung für die LGR-Codierung . . . . .	11.4
<b>4.4c 2009-09-23</b>	
Vorgegebene Unverträglichkeit mit dem <code>inputenx</code> -Paket <b>4.4b 2009-07-24</b>	
Vorgegebene Probleme der Symboldeklaration für die EU1-Codierung . .	11.4
<b>4.4a 2009-07-04</b>	
Hinzugefügte Unterstützung für die EU1-Codierung . . . . .	11.4

Hinzugefügter Zitatstil für Brasilianisch	
Hinzugefügter vorbereitender Zitatstil für Portugiesisch	
<b>4.4 2009-05-30</b>	
Hinzugefügte Paketoption <code>maxlevel</code> . . . . .	2.1
Hinzugefügter spezieller Zitatstil <code>debug</code> . . . . .	10.9
Hinzugefügtes <code>\DeclareQuoteGlyph</code> . . . . .	11.4
Hinzugefügte Unterstützung für OT2 und OT4-Codierungen . . . . .	11.4
Hinzugefügte Unterstützung für T2A-, T2B-, T2C-, X2-Codierungen . . . . .	11.4
Hinzugefügte Unterstützung für die LCY-Codierung . . . . .	11.4
Hinzugefügter Zitatstil für Russisch	
<b>4.3 2008-11-23</b>	
Paketoption <code>babel=tryonce</code> zur Standardeinstellung machen . . . . .	2.1
Interne Updats für das <code>biblatex</code> -Paket	
<b>4.2 2008-06-24</b>	
Upgrade auf <code>etoolbox</code> 1.6	
<b>4.1 2008-04-11</b>	
Vorgegebenes Zeitproblem mit aktiven Zitaten, eingeführt in 4.0	
<b>4.0 2008-03-02</b>	
e-TeX jetzt zwingend erforderlich	
Neue Abhängigkeit vom <code>etoolbox</code> -Paket	
Hinzugefügte Paketoption <code>spanish</code> . . . . .	2.1
Hinzugefügte Variante <code>mexican</code> zum Stil <code>spanish</code> . . . . .	2.1
Entfernte Variante <code>oldstyle</code> vom <code>english</code> -Stil . . . . .	2.1
Entfernte Variante <code>oldstyle</code> vom <code>french</code> -Stil . . . . .	2.1
Entfernte Variante <code>imprimerie</code> vom <code>french</code> -Stil . . . . .	2.1
Erweiterte Dokumentation . . . . .	10.8
Hinzugefügte <code>\openautoquote</code> und <code>\closeautoquote</code> . . . . .	11.5
Hinzugefügte <code>\openinnerquote</code> und <code>\closeinnerquote</code> . . . . .	11.5
Verschobene vordefinierte Stile, Varianten, Optionen auf <code>csquotes.def</code>	
Mehr hinzugefügte Hinweise und Beispiele auf <code>csquotes.cfg</code>	
Hinzugefügte erweiterte PDF-Lesezeichen zum Handbuch	

### 3.8 2008-01-05

Hinzugefügte Variante guillemets* zum Stil swedish . . . . .	2.1
Hinzugefügtes Sprachpseudonym australian . . . . .	8.2
Hinzugefügtes Sprachpseudonym newzealand . . . . .	8.2
Interne Verbesserungen	

### 3.7 2007-03-25

Hinzugefügte Paketooption babel=try . . . . .	2.1
Hinzugefügte Paketooption babel=tryonce . . . . .	2.1
Hinzugefügtes \MakeAutoQuote* . . . . .	4.1
Hinzugefügtes \MakeForeignQuote* . . . . .	4.2
Hinzugefügtes \MakeHyphenQuote* . . . . .	4.2
Hinzugefügtes \mkpretextpunct . . . . .	8.7
Hinzugefügtes \mkpreblockpunct . . . . .	8.7
Hinzugefügtes \mkpreispunct . . . . .	8.7
Entfernter Kompatibilitätscode für den \blockcite-Vermächtnisbefehl	
Interne Updates für das biblatex-Paket	

### 3.6 2006-11-09

Hinzugefügtes \BlockquoteDisable . . . . .	11.3
Festlegung für das amsmath-Paket (aktive Zitate in split und anderer Umgebung)	
Festlegung für das endnotes-Paket (Endnoten in Blockzitaten)	
Überprüfte spanische Zitatstile	

### 3.5 2006-08-24

Ausgetauschte Definitionen von französischen quotes und quotes* Varianten	10.9
Interne Updates für inputenc 1.1 b (2006-05-05)	

### 3.4 2006-04-02

Strengere Validierung von benutzerdefinierten aktiven Zeichen . . . . .	10.3
Autorenschnittstelle jetzt in diesem Handbuch dokumentiert . . . . .	11
Hinzugefügte Dokumentation von \@enablequotes . . . . .	11.1
Hinzugefügte Dokumentation von \@disablequotes . . . . .	11.1

Hinzugefügte Dokumentation von \@verbatimquotes . . . . .	11.1
Hinzugefügte Dokumentation von \@deletequotes . . . . .	11.1
Hinzugefügte Dokumentation in Bezug auf den Umgang mit Strings . . .	11.2
Hinzugefügte Dokumentation der Schnittstelle zu internen Marken . . . .	11.6

### 3.3 2006-02-27

Hinzugefügte Unterstützung für UTF-8-codierte aktive Zitate . . . . .	10.1
Geänderte aktive Zitate, Kategoriecodes 7, 8 nicht länger gültig . . . . .	10.3
Geänderte Trennsymbole, Kategoriecodes 3, 4, 7, 8 nicht länger gültig . .	10.3
Aktive Zitate können jetzt in dem Dokumentkörper definiert werden . . .	4.5
Umbenanntes \RestoreQuotes nach \EnableQuotes . . . . .	4.5
Hinzugefügte \DeleteQuotes . . . . .	4.5
Hinzugefügte \VerbatimQuotes . . . . .	4.5
Hinzugefügte \ExecuteQuoteOptions . . . . .	8.4
Hinzugefügte Paketoption babel=once . . . . .	2.1
Hinzugefügte neue Stilvarianten für französisch . . . . .	10.9
Verbesserte Verschachtelungskontrolle aktiver Blockzitate	
Hergestellte robuste aktive Blockzitate	

### 3.2 2005-12-05

Hinzugefügte Zitatstile für Spanisch  
Vorgegebne Probleme in der hyperref-Schnittstelle

### 3.1 2005-08-29

Hinzugefügtes \textquote . . . . .	3.3
Hinzugefügtes \foreigntextquote . . . . .	3.4
Hinzugefügtes \hyphentextquote . . . . .	3.4
Umbenannte \cquote zu \textcquote . . . . .	5.1
Umbenanntes \foreigncquote zu \foreigntextcquote . . . . .	5.2
Umbenanntes \hyphencquote zu \hyphentextcquote . . . . .	5.2
Erweitertes \textcquote . . . . .	5.1
Erweitertes \foreigntextcquote . . . . .	5.2
Erweitertes \hyphentextcquote . . . . .	5.2
Geänderte Umgebung displayquote . . . . .	6.1

Geänderte Umgebung <code>foreigndisplayquote</code> . . . . .	6.1
Geänderte Umgebung <code>hyphendisplayquote</code> . . . . .	6.1
Erweiterte Umgebung <code>displaycquote</code> . . . . .	6.2
Erweiterte Umgebung <code>foreigndisplaycquote</code> . . . . .	6.2
Erweiterte Umgebung <code>hyphendisplaycquote</code> . . . . .	6.2
Hinzugefügtes <code>\mkmidtextpunct</code> . . . . .	8.7
Hinzugefügtes <code>\mkfintextpunct</code> . . . . .	8.7
Hinzugefügtes <code>\mkmidispunct</code> . . . . .	8.7
Hinzugefügtes <code>\mkfindispunct</code> . . . . .	8.7
Hinzugefügte zusätzliche Umgebung <code>quotetext</code> . . . . .	8.7
Hinzugefügte Erkennung von Absätzen in allen Blockzitatmöglichkeiten .	3.5
<code>\ifquote</code> . . . jetzt verwendbar in <code>\mkcitation</code> und <code>\mkccitation</code> . . . .	8.8
Terminale Zeichensetzung jetzt von allen Zitatmöglichkeiten ausgewertet	
Vermeidung unerwünschter ‘ und ! ‘ Ligaturen in T1-Codierung	
Stellen Sie immer Raumfaktoren für Backendzitate ein.	

### 3.0 2005-07-14

Erweitertes <code>\blockquote</code> . . . . .	3.5
Erweitertes <code>\foreignblockquote</code> . . . . .	3.6
Erweitertes <code>\hyphenblockquote</code> . . . . .	3.6
Erweitertes <code>\setquotestyle</code> . . . . .	3.7
Hinzugefügtes <code>\cquote</code> . . . . .	5.1
Hinzugefügtes <code>\foreigncquote</code> . . . . .	5.2
Hinzugefügtes <code>\hyphencquote</code> . . . . .	5.2
Hinzugefügtes <code>\blockcquote</code> . . . . .	5.3
Hinzugefügtes <code>\foreignblockcquote</code> . . . . .	5.4
Hinzugefügtes <code>\hyphenblockcquote</code> . . . . .	5.4
Hinzugefügte Umgebung <code>displayquote</code> . . . . .	6.1
Hinzugefügte Umgebung <code>foreigndisplayquote</code> . . . . .	6.1
Hinzugefügte Umgebung <code>hyphendisplayquote</code> . . . . .	6.1
Hinzugefügte Umgebung <code>displaycquote</code> . . . . .	6.2
Hinzugefügte Umgebung <code>foreigndisplaycquote</code> . . . . .	6.2

Hinzugefügte Umgebung <code>hyphendisplaycquote</code> . . . . .	6.2
Geändertes <code>\DeclarePlainStyle</code> . . . . .	8.5
Hinzugefügtes <code>\SetCiteCommand</code> . . . . .	8.6
Umbenanntes <code>\blockcite</code> nach <code>\mkcitation</code> . . . . .	8.7
Hinzugefügtes <code>\mkccitation</code> . . . . .	8.7
Hinzugefügtes <code>\mkmidblockpunct</code> . . . . .	8.7
Hinzugefügtes <code>\mkfinblockpunct</code> . . . . .	8.7
Hinzugefügtes <code>\ifquotepunct</code> . . . . .	8.8
Hinzugefügtes <code>\ifquoteterm</code> . . . . .	8.8
Hinzugefügtes <code>\ifquoteperiod</code> . . . . .	8.8
Hinzugefügtes <code>\ifquotecomma</code> . . . . .	8.8
Hinzugefügtes <code>\ifquotesemicolon</code> . . . . .	8.8
Hinzugefügtes <code>\ifquotecolon</code> . . . . .	8.8
Hinzugefügtes <code>\ifquoteexclam</code> . . . . .	8.8
Hinzugefügtes <code>\ifquotequestion</code> . . . . .	8.8
Hinzugefügtes <code>\ifstringblank</code> . . . . .	8.8

Hinzugefügte Auswertung der terminalen Zeichensetzung innerhalb von Blockzitat.

Mit `\nonfrenchspacing` stellen Sie Raumfaktoren für Backendzitate ein.

Verbesserte Verschachtelungskontrolle, wenn e-TeX läuft.