

Das biblatex Paket

Programmierbares Bibliographieren und Zitieren

Philipp Lehman¹
plehman@gmx.net

Version 1.6
29. Juli 2011

Inhaltsverzeichnis

Tabellenverzeichnis	1	3.10. Anwendungshinweise	123
		3.11. Hinweise und Warnungen	135
1. Einleitung	2	4. Autorenanleitung	142
1.1. Über biblatex	2	4.1. Überblick	142
1.2. Lizenz	2	4.2. Bibliographiestile	146
1.3. Rückmeldungen	2	4.3. Zitatstile	159
1.4. Danksagungen	3	4.4. Datenanschluss	162
1.5. Voraussetzungen	3	4.5. Anpassung	172
2. Datenbankübersicht	7	4.6. Hilfsbefehle	179
2.1. Eingabetypen	7	4.7. Zeichensetzung & Abstände	197
2.2. Eingabefelder	15	4.8. Lokalisierte Zeichenketten	205
2.3. Verwendungshinweise	34	4.9. Lokalisierte Module	207
2.4. Hinweise und Vorsichtsmaßnahmen	43	4.10. Formatierungsbefehle	224
		4.11. Hinweise und Warnungen	236
3. Benutzerhandbuch	53	Anhang	254
3.1. Paketoptionen	53	A. Standardsetup von <code>crossref</code>	254
3.2. Globale Anwenderanpassung	75	B. Default Sortierungsschemata	255
3.3. Standardstile	75	B.1. Alphabetisch 1	255
3.4. Sortierungsoptionen	83	B.2. Alphabetisch 2	256
3.5. Bibliographiebefehle	85	B.3. Chronologisch	256
3.6. Zitierungsbefehle	98	C. Versionsgeschichte	257
3.7. Lokalisierungsbefehle	111		
3.8. Formatierungsbefehle	112		
3.9. Anmerkungen zu Sprachen	121		

Tabellenverzeichnis

1. Unterstützte Sprachen	30	5. <code>mcite</code> -artige Befehle	110
2. Datumsspezifikationen	41	6. <code>mcite</code> -artige Syntax	111
3. Kapazität von <code>bibtex8</code>	46	7. Daten-Interface	156
4. Unterstützte Sprachen	57	8. <code>\mkcomprange</code> -Aufbau	194

¹In die deutsche Sprache übersetzt von Christine Römer (Christine_Roemer@t-online.de) und den Studierenden Stephan Wedekind, Lisa Glaser, Maximilian Walter, Franziska Schade, Luise Moder-
sohn, Silvia Müller, Susanne Berghoff, Maiko Sauerteig, Lukas Klimmasch, Peer Aramillo Irizar.
License: \LaTeX Project Public License (LPPL)

1. Einleitung

Dieses Dokument ist ein systematisches Referenzhandbuch für das Paket `biblatex`. Sehen Sie sich auch die Beispieldokumente an, die mit diesem Paket mitgeliefert wurden, um einen ersten Eindruck zu bekommen.² Für einen schnellen Einstieg, lesen Sie §§ 1.1, 2.1, 2.2, 2.3, 3.1, 3.3, 3.5, 3.6, 3.10.

1.1. Über `biblatex`

Dieses Paket enthält bibliographische Möglichkeiten für die Verwendung von LaTeX zusammen mit BibTeX. Das Paket ist eine komplette Neuimplementierung der bibliographischen Einrichtungen, welche von LaTeX zur Verfügung gestellt werden. Es erweitert die Möglichkeiten, in welcher Art und Weise LaTeX mit BibTeX interagiert auf eine fundamentale Weise. Mit `biblatex` wird BibTeX nur verwendet, um die Bibliographie zu sortieren und Labels zu erstellen. Anstatt in `bst`-Dateien umgesetzt zu werden, wird die Formatierung der Bibliographie vollständig von LaTeX-Makros gesteuert, daher der Name `biblatex`. Gute Kenntnisse in LaTeX sollten ausreichend sein, um eine neue Bibliographie und Zitatvorlage zu erstellen. Es besteht keine Notwendigkeit die „postfix-stack-Sprache“ von BibTeX zu lernen. Dieses Paket unterstützt auch unterteilte Bibliographien, mehrere Bibliographien innerhalb eines Dokuments und separate Listen von bibliographischen Kürzeln. Bibliographien können in Teile unterteilt und/oder nach Themen segmentiert werden. Genau wie die Bibliographiestile, können alle Zitatbefehle frei definiert werden. Das Paket ist vollständig übersetzt und kann mit dem Paket `babel` benutzt werden. Bitte siehe Tabelle 1 für eine Liste der Sprachen, die derzeit von diesem Paket unterstützt werden.

1.2. Lizenz

Copyright © 2006–2011 Philipp Lehman. Es ist erlaubt, zu kopieren, zu verteilen und /oder die Software zu modifizieren, unter den Bedingungen der LaTeX Project Public License, Version 1.3.1.³ Dieses Paket wird vom Autor weiter betreut.

1.3. Rückmeldungen

Bitte benutzen Sie die `biblatex`-Projektseite auf SourceForge um Fehler zu melden und um Featureanfragen zu senden.⁴ Es gibt zwei Reiter auf dieser Seite: Bugs für Fehlermeldungen und Features für Featureanfragen.⁵ Wählen Sie einen Reiter, klicken auf „Add new“, um die Anfrage oder den Bericht zu senden. Ich bin bereit,

²[\biblatexctan/doc/examples](#)

³<http://www.ctan.org/tex-archive/macros/latex/base/lppl.txt>

⁴<http://sourceforge.net/projects/biblatex/>

⁵http://sourceforge.net/tracker2/?group_id=244752

das neue Feature einzubauen, wenn die Anfrage sinnvoll und relevant für mehrere Nutzer ist und mit der Architektur und Philosophie von `biblatex` übereinstimmt. Bevor Sie eine Featureanfrage senden, stellen Sie bitte sicher, das Sie dieses Benutzerhandbuch gründlich gelesen haben. Wenn Sie keinen Bug melden wollen oder eine Featureanfrage stellen, sondern einfach Hilfe benötigen, können Sie Ihre Frage in der `comp.text.tex` Newsgroup stellen. Ich kann keine Endnutzer-Unterstützung per E-Mail oder über die Projektseite bieten.

1.4. Danksagungen

Die Sprachmodule dieses Pakets beinhalten die Beiträge von (in der Reihenfolge der Einreichung): Ignacio Fernández Galván (Spanisch), Enrico Gregorio (Italienisch), Johannes Wilm (Dänisch/Norwegisch), Per Starbäck, Carl-Gustav Werner, Filip Åsblom (Schwedisch), Augusto Ritter StoVel (Brasilianisch), Alexander van Loon (Niederländisch), Apostolos Syropoulos (Griechisch), Hannu Väisänen (Finnisch), Prokopis (Griechisch), Mateus Araújo (Brasilianisch), Andrea Marchitelli (Italienisch), José Carlos Santos (Portugiesisch).

1.5. Voraussetzungen

Dieser Abschnitt gibt einen Überblick über alle nötigen Mittel, die für dieses Paket erforderlich sind und diskutiert Kompatibilitätsprobleme.

1.5.1. Anforderungen

Die Mittel, die in diesem Abschnitt aufgeführt sind, sind unbedingt erforderlich, damit `biblatex` funktionieren kann. Das Paket wird nicht funktionieren, wenn sie nicht verfügbar sind.

e-TeX Das `biblatex`-Paket erfordert e-TeX. TeX-Distributionen enthalten seit geraumer Zeit e-TeX-Binärdateien, die bekannten Distributionen nutzen diese heutzutage standardmäßig. Das `biblatex`-Paketes überprüft, ob es unter e-TeX funktioniert. Wenn Sie Ihre Dokumente wie gewohnt kompilieren, stehen die Chancen gut, dass es funktioniert. Wenn Sie eine Fehlermeldung erhalten, versuchen Sie Ihr Dokument mit `elatex` statt `latex` bzw. `pdfelatex` statt `pdflatex` zu kompilieren.

BibTeX Das `biblatex`-Paket verwendet standardmäßig BibTeX als Datenbankhintergrund. Während es für ältere BibTeX Binärdateien `biblatex` ausreicht, wird die Nutzung von `bibtex8` empfohlen. Wenn Ihre TeX-Distribution kein `bibtex8` enthält, können Sie es von CTAN erhalten.⁶

⁶<http://www.ctan.org/tex-archive/biblio/bibtex/8-bit/>

- Biber** Biber ist die neuste Generation des `biblatex`-Hintergrundes. Es ist derzeit in Entwicklung und kann aber bereits statt BibTeX verwendet werden. Sie benötigen nur einen Hintergrund, entweder BibTeX oder Biber. Biber ist über SourceForge erhältlich.⁷ Seit 2011 ist es auch in den wichtigen Distributionen (wie TeX Live) enthalten.
- etoolbox** Dieses LaTeX-Paket, das automatisch geladen wird, bietet generische Programmierung, welche zur Einrichtungen von `biblatex` erforderlich ist. Es ist über CTAN⁸ erhältlich.
- logreq** Dieses LaTeX-Paket, das auch automatisch geladen wird, stellt ein Programm für das Schreiben von maschinenlesbaren Nachrichten für eine zusätzliche Log-Datei bereit. Es ist erhältlich über CTAN.⁹

Abgesehen von den oben genannten Mitteln, erfordert `biblatex` auch die Standard-LaTeX-Pakete `keyval` und `ifthen` sowie das `url`-Paket. Diese Pakete sind in allen gängigen TeX-Distributionen enthalten und werden automatisch geladen.

1.5.2. Empfohlene Pakete

Die Pakete in diesem Abschnitt sind nicht erforderlich für die Funktionsfähigkeit von `biblatex`, aber sie bieten zusätzliche Funktionen oder Verbesserungen zu vorhandenen Features. Die Reihenfolge, wie man die Pakete lädt, ist unerheblich.

- babel** Das `babel`-Paket enthält die Kernarchitektur für mehrsprachigen Satz. Wenn Sie in einer anderen Sprache als in amerikanisches Englisch schreiben, wird die Nutzung dieses Paketes empfohlen. Wenn `biblatex` geladen wird, wird `babel` automatisch erkannt.
- csquotes** Wenn dieses Paket verfügbar ist, wird `biblatex` seinen Sprachteil nutzen, um bestimmte Titel in Anführungszeichen zu setzen. Wenn nicht, nutzt `biblatex` passende Zitierweisen für das amerikanische Englisch. Beim Schreiben in einer anderen Sprache, wird das Laden von `csquotes` empfohlen.¹⁰

1.5.3. Kompatible Klassen und Pakete

Das Paket `biblatex` bietet Kompatibilitätscodes für Klassen und Pakete, welche in diesem Abschnitt aufgeführt sind.

- hyperref** Das `hyperref`-Paket verwandelt Zitierungen in Hyperlinks. Siehe `hyperref`- und `backref`-Paketooptionen in § 3.1.2.1 für weitere Details. Bei der Verwendung des `hyperref` Paketes, wird empfohlen, vorher `biblatex` zu laden.

⁷<http://biblatex-biber.sourceforge.net/>

⁸<http://www.ctan.org/tex-archive/macros/latex/contrib/etoolbox/>

⁹<http://www.ctan.org/tex-archive/macros/latex/contrib/logreq/>

¹⁰<http://www.ctan.org/tex-archive/macros/latex/contrib/csquotes/>

showkeys Das `showkeys`-Paket zeigt die internen Schlüssel, unter anderem von Zitaten im Text und Begriffen in der Bibliographie. Die Reihenfolge, wie man die Pakete lädt, ist unerheblich.

memoir Bei Verwendung der `memoir`-Klasse sind die Standard-Bibliographie-Überschriften so angepasst, dass sie gut mit dem Standard-Layout dieser Klasse zusammenpassen. Siehe § 3.11.2 für weitere Verwendungshinweise.

KOMA-Script Bei Verwendung der `scrartcl`-, `scrbook`- oder `scrreprt`-Klassen, die Standard-Bibliographie-Überschriften werden so angepasst, dass sie gut mit dem Standard-Layout dieser Klassen zusammenpassen. Siehe § 3.11.1 für die weitere Verwendungshinweise.

1.5.4. Inkompatible Pakete

Die Pakete in diesem Abschnitt sind nicht kompatibel mit `biblatex`. Seit der Reimplementierung der Bibliographieausstattung von LaTeX hat `biblatex` Probleme mit allen Paketen, die die selben Ausstattungen modifiziert haben. Dies ist nicht spezifisch von `biblatex`. Einige der unten aufgeführten Pakete sind aus dem selben Grund miteinander inkompatibel.

babelbib Das `babelbib`-Paket bietet Unterstützung für mehrsprachige Bibliographien. Dies ist ein Standardfeature von `biblatex`. Verwenden Sie das Feld `hyphenation` und die Paketoption `babel` für ähnliche Funktionen. Beachten Sie, dass `biblatex` automatisch die Sprache entsprechend dem Hauptdokument anpasst, wenn `babel` geladen ist. Sie brauchen nur die oben genannten Funktionen, falls Sie die Sprachen auf einer pro-Eintrag-Grundlage innerhalb der Bibliographie ändern wollen. Siehe §§ 2.2.3 and 3.1.2.1 für Details. Siehe auch § 3.7.

backref Das `backref` Paket schafft Rückverweise in die Bibliographie. Siehe die Paketoptionen `hyperref` und `backref` in § 3.1.2.1 für vergleichbare Funktionalität.

bibtopic Das `bibtopic`-Paket bietet Unterstützung für Bibliographien gegliedert nach Themen, Typ oder anderen Kriterien. Für Bibliographien, unterteilt nach Themen, finden Sie in der Kategorie `Feature` (in § 3.5.6) und die entsprechenden Filter in § 3.5.2. Alternativ können Sie das Feld `keywords` in Verbindung mit den `keyword` und `notkeyword` Filter für vergleichbarer Funktionalität verwenden, vgl. §§ 2.2.3 and 3.5.2 für weitere Einzelheiten. Für Bibliographien unterteilt nach Typ, nutzen sie den `type` und `nottype` Filtern. Siehe auch § 3.10.4 für Beispiele.

bibunits Das `bibunits`-Paket bietet Unterstützung für mehrere Teilbibliographien (z. B. pro Kapitel). Siehe `chapterbib`.

- chapterbib** Das `chapterbib`-Paket bietet Unterstützung für mehrere Teilbibliographien. Verwenden Sie die `refsection`-Umgebung und den `section`-Filter für vergleichbare Funktionalität. Alternativ können Sie auch die `refsegment`-Umgebung und den `segment`-Filter verwenden. Siehe §§ 3.5.4, 3.5.5, 3.5.2 für weitere Einzelheiten. Siehe auch § 3.10.3 für Beispiele.
- cite** Das `cite`-Paket sortiert automatisch die numerischen Zitate und kann eine Liste von komprimierten fortlaufenden Nummern auf einen Bereich erstellen. Es macht auch die Interpunktion, welche in Zitaten verwendet wird, konfigurierbar. Für sortierte und komprimierte numerische Zitate finden Sie in der `sortcites`-Paketoption in § 3.1.2.1 und den `numeric-comp`-Zitierstil in § 3.3.1. Für konfigurierbare Interpunktion siehe § 3.8.
- citeref** Ein weiteres Paket für die Erstellung für Rückverweise in die Bibliographie. Siehe auch `backref`.
- inlinebib** Das `inlinebib`-Paket ist für traditionelle Zitate in Fußnoten konzipiert. Für vergleichbare Funktionalität finden Sie den ausführlichen Zitierstil in § 3.3.1.
- jurabib** Ursprünglich für Zitierungen im Jurastudium und (meist deutsche) gerichtliche Dokumente entworfen, bietet das `jurabib`-Paket auch Funktionen an für Anwender in den Geisteswissenschaften. Im Hinblick auf die Eigenschaften gibt es einige Ähnlichkeiten zwischen `jurabib` und `biblatex`, aber die Ansätze der beiden Pakete sind recht verschieden. Seit die beiden Pakete `jurabib` und `biblatex` voll ausgestattet sind, ist die Liste der Gemeinsamkeiten und Unterschiede zu lang, um hier dargestellt zu werden.
- mcite** Das `mcite`-Paket bietet Unterstützung für gruppierte Zitate, d. h. es können mehrere Elemente als eine einzige Referenz zitiert und als einzelner Eintrag in der Bibliographie aufgelistet werden. Die Zitatgruppen sind so definiert, wie die Elemente zitiert wurden. Dies funktioniert nur mit unsortierten Bibliographien. Das Paket `biblatex` unterstützt auch gruppierte Zitate, welche in diesem Handbuch ‚Eintragstypen/entry sets‘ oder ‚Referenztypen/reference sets‘ heißen. Siehe §§ 3.10.5, 3.5.10, 3.6.10 für Einzelheiten.
- mciteplus** Eine deutlich verbesserte Neuimplementierung des `mcite`-Paketes, die Gruppierung in sortierten Bibliographien unterstützt. Siehe `mcite`.
- multibib** Das `multibib`-Paket bietet Unterstützung für Bibliographien, welche thematisch oder nach anderen Kriterien unterteilt sind. Siehe `bibtopic`.
- natbib** Das `natbib`-Paket unterstützt numerische und Autor-Jahr-Zitat-Systeme unter Einbeziehung von Sortier- und Kompressions-Codes, welche Sie im `cite`-Paket finden. Es bietet auch zusätzliche Zitierbefehle und mehrere Konfigurationsoptionen. Sehen Sie die `numeric` und `author-year`-Zitierstile und ihre Varianten in § 3.3.1, die

sortcites-Paket-Option in § 3.1.2.1, die Zitierbefehle in § 3.6 und die Einstellungen in den §§ 3.5.7, 3.5.8, 3.8 für vergleichbare Funktionalität. Siehe auch § 3.6.9.

splitbib Das `splitbib`-Paket bietet Unterstützung für Bibliographien, welche thematisch unterteilt sind. Siehe `bibtopic`.

ucs Das `ucs`-Paket bietet Unterstützung für UTF-8-kodierte Eingaben. Entweder verwenden sie von `inputenc` das Standard-`utf8`-Modul oder eine unicode-fähige Engine wie XeTeX oder LuaTeX.

2. Datenbankübersicht

Dieser Teil des Handbuchs dokumentiert die BibTeX-Schnittstelle des Paketes. Beachten sie, dass Sie `biblatex` nicht in Verbindung mit beliebigen `bst`-Dateien verwenden können, da das Paket an einer besonderen BibTeX-Schnittstelle hängt.

2.1. Eingabetypen

Dieser Abschnitt gibt einen Überblick über die Eingabetypen und deren Felder, welche von `biblatex` unterstützt werden.

2.1.1. Grundtypen

Die folgende Liste enthält die Felder, die von jeweiligen Eingabetyp unterstützt werden. Beachten Sie, dass die Zuordnung der Felder zu einem Eingabetyp im Ermessen des Bibliographiestils liegt. Die folgende Liste verfolgt daher zwei Ziele. Sie zeigt die Felder, welche von den Standardstils unterstützt werden, die in diesem Paket enthalten sind und sie dienen auch als Vorbild für eigene Designs. Beachten Sie, dass die ‚erforderlichen‘ Felder nicht in allen Fällen unbedingt erforderlich sind, siehe § 2.3.2 für weitere Einzelheiten. Die Felder ‚optional‘ sind optional in einem technischen Sinne. Bibliographische Formatierungsregeln erfordern in der Regel mehr als nur die ‚erforderlich‘-Felder. Die Standardstile unterliegen in der Regel keinen formalen Gültigkeitskontrollen, haben aber jedoch eigene Designs. Generische Felder wie `abstract` und `annotation` oder `label` und `shorthand` sind nicht in der Liste unten aufgeführt, weil sie unabhängig vom Eingabetyp sind. Die speziellen Felder, die in § 2.2.3 erklärt werden und auch unabhängig vom Eingabetyp sind, sind ebenfalls nicht in der Liste enthalten.

article Ein Artikel in einem Journal, einer Zeitschrift, einer Zeitung oder einem anderem Periodikum, welches einen eigenen Abschnitt mit einem eigenen Titel erhält. Der Titel des Periodikums ist im `journaltitle`-Feld enthalten. Wenn die Ausgabe neben dem Haupttitel des Periodikums einen eigenen Titel hat, ist dieser im `issuetitle`-Feld

einzugeben. Beachten Sie, dass `editor` und verwandte Felder zum Journal gehören, während `translator` und verwandte Felder, zu `@article` gehören.

Pflichtfelder: `author`, `title`, `journaltitle`, `year/date`

Wahlfelder: `translator`, `annotator`, `commentator`, `subtitle`, `titleaddon`, `editor`, `editora`, `editorb`, `editorc`, `journalsubtitle`, `issuetitle`, `issue-subtitle`, `language`, `origlanguage`, `series`, `volume`, `number`, `eid`, `issue`, `month`, `pages`, `version`, `note`, `issn`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`

book Ein Buch mit einem oder mehreren Autoren, wo die Autoren alle zusammen gearbeitet haben. Beachten Sie, dass dieser Eingabetyp auch die Funktion des `@inbook`-Typs des herkömmlichen BibTeX umfasst, siehe § 2.3.1 für weitere Einzelheiten.

Pflichtfelder: `author`, `title`, `year/date`.

Wahlfelder: `editor`, `editora`, `editorb`, `editorc`, `translator`, `annotator`, `commentator`, `introduction`, `foreword`, `afterword`, `subtitle`, `titleaddon`, `maintitle`, `mainsubtitle`, `maintitleaddon`, `language`, `origlanguage`, `volume`, `part`, `edition`, `volumes`, `series`, `number`, `note`, `publisher`, `location`, `isbn`, `chapter`, `pages`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`

mvbook Ein mehrbändiges Buch. Für rückwärtskompatible mehrbändige Bücher gilt der Support des Eintragsstypes `@book`. Jedoch ist es ratsam, den Eintragstyp `@mvbook` zu nehmen.

Pflichtfelder: `author`, `title`, `year/date` Wahlfelder: `editor`, `editora`, `editorb`, `editorc`, `translator`, `annotator`, `commentator`, `introduction`, `foreword`, `afterword`, `subtitle`, `titleaddon`, `language`, `origlanguage`, `edition`, `volumes`, `series`, `number`, `note`, `publisher`, `location`, `isbn`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`

inbook Ein Teil eines Buches, der eine in sich geschlossene Einheit mit seinem eigenen Titel bildet. Beachten Sie, dass das Profil dieses Eingabetyps verschieden ist vom Standard-BibTeX, siehe § 2.3.1.

Pflichtfelder: `author`, `title`, `booktitle`, `year/date`

Wahlfelder: `bookauthor`, `editor`, `editora`, `editorb`, `editorc`, `translator`, `annotator`, `commentator`, `introduction`, `foreword`, `afterword`, `subtitle`, `titleaddon`, `maintitle`, `mainsubtitle`, `maintitleaddon`, `booksubtitle`, `booktitleaddon`, `language`, `origlanguage`, `volume`, `part`, `edition`, `volumes`, `series`, `number`, `note`, `publisher`, `location`, `isbn`, `chapter`, `pages`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`

- bookinbook** Ähnlich wie `@inbook`, aber bestimmt für Artikel, welche ursprünglich als eigenständiges Buch veröffentlicht werden sollten. Ein typisches Beispiel sind Bücher, welche in einem Sammelwerk der Autors abgedruckt sind.
- suppbook** Zusätzliches Material in einem `@book`. Diese Typ ist eng mit dem `@inbook`-Eingabetyp verwandt. Während `@inbook` in erster Linie für einen Teil eines Buches mit eigenem Titel bestimmt ist (d. h. einem einzelnen Aufsatz in einer Sammlung von Essays von demselben Autor), ist diese Art vorgesehenen für Elemente wie Vorreden, Einleitungen, Vorworte, Nachworte, etc., welche oft nur eine allgemeine Bezeichnung haben. Style Guides erfordern solche Elemente, um andere `@inbook`-Elemente zu formatiert. Die Standardstile behandeln diese Eingabetypen ähnlich wie `@inbook`.
- booklet** Eine buchähnliche Arbeit ohne formalen Verlag oder Sponsoring Institution. Verwenden Sie das Feld `howpublished` für zusätzliche Veröffentlichungsinformationen, wenn zutreffend. Das Feld `type` kann auch sinnvoll sein.
- Pflichtfelder: `author/editor`, `title`, `year/date`
 Wahlfelder: `subtitle`, `titleaddon`, `language`, `howpublished`, `type`, `note`, `location`, `chapter`, `pages`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`
- collection** Ein Buch mit mehreren, in sich geschlossenen Beiträgen, die verschiedene Autoren haben und ihre eigenen Titel. Die Arbeit als Ganzes hat keinen allgemeinen Autor, aber es hat in der Regel einen Herausgeber.
- Pflichtfelder: `editor`, `title`, `year/date`
 Wahlfelder: `editora`, `editorb`, `editorc`, `translator`, `annotator`, `commentator`, `introduction`, `foreword`, `afterword`, `subtitle`, `titleaddon`, `maintitle`, `main-subtitle`, `maintitleaddon`, `language`, `origlanguage`, `volume`, `part`, `edition`, `volumes`, `series`, `number`, `note`, `publisher`, `location`, `isbn`, `chapter`, `pages`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`
- mvcollection** Eine mehrbändige Kollektion (`@collection`). Rückwärtskompatible mehrbändige Collections werden unterstützt vom Eintragstyp `@collection`. Trotzdem ist es sinnvoll, den Eintragstyp `@mvcollection` zu nehmen.
- Pflichtfelder: `editor`, `title`, `year/date`. Wahlfelder: `editora`, `editorb`, `editorc`, `translator`, `annotator`, `commentator`, `introduction`, `foreword`, `afterword`, `subtitle`, `titleaddon`, `language`, `origlanguage`, `edition`, `volumes`, `series`, `number`, `note`, `publisher`, `location`, `isbn`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`.
- incollection** Ein Beitrag zu einer Sammlung, die eine in sich geschlossene Einheit bildet mit einem deutlichen Autor und Titel. Der `author` bezieht sich auf den `title`, der `editor` auf `booktitle`, d. h., den Titel der Sammlung.

Pflichtfelder: author, editor, title, booktitle, year/date

Wahlfelder: editora, editorb, editorc, translator, annotator, commentator, introduction, foreword, afterword, subtitle, titleaddon, maintitle, main-subtitle, maintitleaddon, booksubtitle, booktitleaddon, language, orig-language, volume, part, edition, volumes, series, number, note, publisher, location, isbn, chapter, pages, addendum, pubstate, doi, eprint, eprint-class, eprinttype, url, urldate

suppcollection Zusätzliches Material in @collection. Dieser Typ ist ähnlich wie @collection, aber bezieht sich auf den @collection-Eingabetyp. Der Standardstil behandelt diesen Eingabetyp ähnlich wie @incollection.

manual Technische oder sonstige Unterlagen, nicht unbedingt in gedruckter Form. Der author oder editor sind im Sinne des § 2.3.2 weggelassen worden.

Pflichtfelder: author/editor, title, year/date

Wahlfelder: subtitle, titleaddon, language, edition, type, series, number, version, note, organization, publisher, location, isbn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate

misc Ein Typ für Einträge, die in keine andere Kategorie passen. Verwenden Sie das Feld howpublished für zusätzliche Veröffentlichungsinformationen, wenn zutreffend. Das Feld type kann auch sinnvoll sein. Der author, editor und year sind im Sinne des § 2.3.2 weggelassen worden.

Pflichtfelder: author/editor, title, year/date

Wahlfelder: subtitle, titleaddon, language, howpublished, type, version, note, organization, location, date, month, year, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate

online Eine Online-Quelle. author, editor, und year sind im Sinne des § 2.3.2 weggelassen worden. Dieser Eingabetyp ist für reine Onlinequellen wie Websites gedacht. Beachten Sie, dass alle Eingabetypen das url-Feld unterstützen. Zum Beispiel, wenn Sie einen Artikel aus einer Zeitschrift, die online verfügbar ist, zitieren, verwenden Sie den @article-Typ und sein url-Feld.

Pflichtfelder: author/editor, title, year/date, url

Wahlfelder: subtitle, titleaddon, language, version, note, organization, date, month, year, addendum, pubstate, urldate

patent Ein Patent oder eine Patentanfrage. Die Anzahl oder Eintragsnummer ist im Feld number angegeben. Verwenden Sie das Feld type, um den Typ zu spezifizieren und das Feld location, um den Umfang des Patentbesitzes anzugeben, falls dieser vom Standardmäßigen abweicht. Beachten Sie, dass das location-Feld eine Schlüsselliste für diesen Eingabetyp ist, siehe § 2.2.1 für weitere Einzelheiten.

Pflichtfelder: author, title, number, year/date

Wahlfelder: holder, subtitle, titleaddon, type, version, location, note, date, month, year, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate

periodical Eine ganze Ausgabe einer Zeitschrift, ähnlich einer speziellen Ausgabe einer Zeitschrift. Der Titel der Zeitschrift ist im Feld `title` gegeben. Wenn die Ausgabe zusätzlich zu dem Haupttitel der Zeitschrift einen eigenen Titel hat, wird dieser im Feld `issuetitle` angegeben. Der `editor` wurde im Sinne von § 2.3.2 weggelassen.

Pflichtfelder: editor, title, year/date

Wahlfelder: editora, editorb, editorc, subtitle, issuetitle, issuesubtitle, language, series, volume, number, issue, date, month, year, note, issn, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate

supperiodical Zusätzliches Material in einem `@periodical`. Dieser Typ ist ähnlich wie `@suppbook`, gehört aber zum `@periodical`-Eingabetyp. Die Rolle dieses Eintragstyps kann besser verstanden werden, wenn man bedenkt, dass der `@article`-Typ auch `@inperiodical` genannt werden könnte. Dieser Typ kann nützlich sein, wenn es um Dinge wie regelmäßige Kolumnen, Nachrufe, Leserbriefe, etc. geht, die nur eine allgemeine Bezeichnung haben. Stile erfordern solche Elemente, da sie anders als Artikel formatiert sind. Der Standardstil behandelt diesen Eingabetyp ähnlich wie `@article`.

proceedings Bericht von einer Konferenz. Dieser Typ ist ähnlich wie `@collection`. Er unterstützt ein optionales Feld `organization`, welches das Sponsorinstitut beinhaltet. Der `editor` wurde im Sinne von § 2.3.2 weggelassen.

Pflichtfelder: editor, title, year/date.

Wahlfelder: subtitle, titleaddon, maintitle, mainsubtitle, maintitleaddon, eventtitle, eventdate, venue, language, volume, part, volumes, series, number, note, organization, publisher, location, month, isbn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

mvproceedings Ein mehrbändiger Bericht von einer Konferenz (`@proceedings`). Rückwärtskompatible mehrbändige Proceedings werden unterstützt vom Eintragstyp `@proceedings`. Trotzdem ist es sinnvoll, den Eintragstyp `@mvproceedings`.

Pflichtfelder: editor, title, year/date.

Wahlfelder: subtitle, titleaddon, eventtitle, eventdate, venue, language, volumes, series, number, note, organization, publisher, location, month, isbn, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

inproceedings Ein Artikel in einem Konferenzband. Dieser Typ ist ähnlich wie `@incollection`. Er unterstützt ein optionales Feld `organization`.

Pflichtfelder: author, editor, title, booktitle, year/date

Wahlfelder: subtitle, titleaddon, maintitle, mainsubtitle, maintitleaddon, booksubtitle, booktitleaddon, eventtitle, eventdate, venue, language, volume, part, volumes, series, number, note, organization, publisher, location, month, isbn, chapter, pages, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate

- reference** Ein Nachschlagewerk wie Lexika und Wörterbücher. Dies ist eine spezifischere Variante des allgemeinen @collection-Eintragstyps. Der Standardstil behandelt diesen Eingabetyp ähnlich wie @collection.
- mvreference** Ein mehrbändiges Nachschlagewerk (@reference). Die Standardstile behandeln diesen Eintragstyp als Alias von @mvcollection. Rückwärtskompatible mehrbändige References werden unterstützt vom Eintragstyp @reference. Trotzdem ist es sinnvoll, den Eintragstyp @mvreference zu nehmen.
- inreference** Artikel in einem Nachschlagewerk. Dies ist eine spezifischere Variante des allgemeinen @incollection-Eintragstyps. Der Standardstil behandelt diesen Eingabetyp ähnlich wie @incollection.
- report** Ein technischer Bericht, Forschungsbericht oder ein Weißbuch, wurde von einer Universität oder einer anderen Institution veröffentlicht. Verwenden Sie das Feld type, um die Art des Berichts angeben. Das Sponsorinstitut wird im Feld institution angegeben.
- Pflichtfelder: author, title, type, institution, year/date
 Wahlfelder: subtitle, titleaddon, language, number, version, note, location, month, isrn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate
- set** Eine Eintragsammlung. Dieser Eintrag Typ ist etwas Besonderes, siehe § 3.10.5 für Details.
- thesis** Eine Doktorarbeit, die für eine Bildungseinrichtung geschrieben wurde, als Voraussetzung für einen wissenschaftlichen Grad. Verwenden Sie das Feld type, um die Art der Arbeit angeben.
- Pflichtfelder: author, title, type, institution, year/date
 Wahlfelder: subtitle, titleaddon, language, note, location, month, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate
- unpublished** Eine Arbeit mit einem Autor und einen Titel, die bisher nicht offiziell, ein Manuskript oder das Skript eines Vortrags, veröffentlicht wurde. Verwenden Sie die Felder howpublished und note, um zusätzliche Angaben zu machen.

Pflichtfelder: `author`, `title`, `year/date`

Wahlfelder: `subtitle`, `titleaddon`, `language`, `howpublished`, `note`, `location`, `date`, `month`, `year`, `addendum`, `pubstate`, `url`, `urldate`

`custom[a–f]` Benutzerdefinierte Typen für spezielle Bibliographiestile. Diese werden nicht von den Standardstilen verwendet.

2.1.2. Typ-Aliasnamen

Die Eintragsarten, die in diesem Abschnitt aufgeführt werden, sind für Abwärtskompatibilität mit herkömmlichen BibTeX-Stiles bereitgestellt. Diese Aliase werden von BibTeX aufgelöst, wenn die Daten exportiert werden. Die Bibliographiestile verweisen mit ihren Einträgen auf den Alias und nicht auf den Aliasnamen. Alle unbekannt Eintragsarten sind in der Regel als `@misc` exportiert.

`conference` Ein älteres Alias `@inproceedings`.

`electronic` Ein Alias `@online`.

`mastersthesis` Ähnlich wie `@thesis`, außer dass das Feld `type` optional ist und standardmäßig auf den lokalisierten Begriff ‚Master’s thesis‘ gesetzt ist. Sie können immer noch das Feld `type` nutzen, um dies zu überschreiben.

`phdthesis` Ähnlich wie `@thesis`, außer dass das Feld `type` optional ist und standardmäßig auf den lokalisierten Begriff ‚PhD thesis‘ gesetzt ist. Sie können immer noch das Feld `type` nutzen, um dies zu überschreiben.

`techreport` Ähnlich wie `@report` mit der Ausnahme, dass das Feld `type` optional ist und standardmäßig auf den lokalisierten Begriff ‚technical report‘ gesetzt ist. Sie können immer noch das Feld `type` zu überschreiben.

`www` Ein Alias für `@online`, für `jurabib`-Kompatibilität bereitgestellt.

2.1.3. Nicht unterstützte Typen

Die Typen in diesem Abschnitt sind ähnlich wie die benutzerdefinierte Typen `@customa` bis `@customf`, d. h., sie werden von der `bib`-Datei unter ihrem richtigen Namen anstatt als `@misc`-Einträge exportiert. Dies kann in benutzerdefinierten Designs Bibliographien nützlich sein. Die Standard Bibliographie Designs bieten keine spezielle Unterstützung für diese Typen. Bei Verwendung der Standardstilen werden sie entweder ignoriert oder als Aliase eines verwandten allgemeineren Typs behandelt (siehe unten für Details).

`artwork` Werke der bildenden Kunst wie Gemälde, Skulpturen und Installationen. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.

- audio** Audio-Aufnahmen, in der Regel auf Audio-CD, DVD, Audio-Kassette oder ähnlichen Medien. Diese Typ wird derzeit von der Standardstilen nicht unterstützt. Siehe auch `@music`.
- bibnote** Ein spezieller Eingabetyp, der nicht wie andere in `bib`-Dateien verwendet wird. Er ist vorgesehenen für Pakete von Drittanbietern wie `notes2bib`, welche Anmerkungen zu der Bibliographie hinzufügen. Die Anmerkungen sollen in das Feld `note`. Es wird darauf hingewiesen, dass der `@bibnote`-Typ in keinem Zusammenhang zum `\defbibnote`-Befehl steht. `\defbibnote` ist für das Hinzufügen von Kommentaren zu Beginn oder am Ende der Bibliographie, während der `@bibnote`-Typ für Pakete gedacht ist, die Anmerkungen zu den Bibliographie-Einträgen hinzufügen.
- commentary** Kommentare, die einen anderen Status als normale Bücher, wie Rechts- Kommentare, haben. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- image** Bilder, Gemälde, Fotografien und ähnlichen Medien. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- jurisdiction** Gerichtsurteile, gerichtliche Aufnahmen und ähnliche Dinge. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- legislation** Gesetze, Gesetzesvorlage, Legislativvorschläge und ähnliche Dinge. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- legal** Juristische Dokumente wie Verträge. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- letter** Persönliche Korrespondenz wie Briefe, E-Mails, Memoiren , etc. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- movie** Bewegte Bilder. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt. Siehe auch `@video`.
- music** Musikalische Aufnahmen. Dies ist eine speziellere Variante von `@audio`. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- performance** Musik- und Theateraufführungen sowie andere Werke der darstellenden Künste. Dieser Typ bezieht sich speziell auf die Veranstaltung als eine Aufnahme, eine Partitur oder eine gedruckte Aufführung. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- review** Kritik zu anderen Arbeiten. Dies ist eine spezielle Variante des `@article`-Types. Die Standardstile behandeln diesen Eintrag wie `@article`.
- software** Computersoftware. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.

- standard** Nationale und internationale Standards durch ein zuständiges Gremium wie die Internationale Organisation für Standardisierung. Dieser Typ wird derzeit von den Standardstilen nicht unterstützt.
- video** Audiovisuelle Aufzeichnungen, in der Regel auf DVD, VHS-Kassette oder ähnlichen Medien. Dieser Typ wird derzeit von Standardstilen nicht unterstützt. Siehe auch `@movie`.

2.2. Eingabefelder

Dieser Abschnitt gibt einen Überblick über die Eingabefelder, die von `biblatex` unterstützt werden. Siehe § 2.2.1 für eine Einführung in die Datentypen, die von diesem Paket unterstützt werden und §§ 2.2.2 und 2.2.3 für die eigentliche Felderaufstellung.

2.2.1. Datentypen

In `bib`-Dateien werden alle bibliographischen Daten in Feldern angegeben. Einige dieser Felder, zum Beispiel `author` und `editor`, können eine Liste von Elementen enthalten. Diese Listenstruktur wird in BibTeX mit Hilfe des Stichwortes `,and` erreicht, welches zur Trennung der einzelnen Elemente in der Liste gebraucht wird. Das `biblatex`-Paket beinhaltet drei verschiedene Datenarten, um bibliographische Daten zu handhaben: Namenslisten, Wörterlisten und Felder. Es gibt auch mehrere Listen und Feld-Subtypen. Dieser Abschnitt gibt einen Überblick über die Datentypen, welche durch dieses Paket unterstützt werden. Siehe §§ 2.2.2 und 2.2.3 für Informationen über die Umwandlung von BibTeX-Feldern in `biblatex`-Datentypen.

Namenslisten werden analysiert und in die einzelnen Positionen und das Bindewort `and` gespalten. Jedes Element in der Liste wird dann in vier Namensbestandteile zerlegt: Vorname, Namenspräfix (von, van, of, da, de, della, ...), Nachnamen und Namenssuffix (Junior, Senior, ...). Namentliche Listen können in der `bib`-Datei mit dem Schlüsselwort `,and others` gekürzt werden. Typische Beispiele von Namenslisten sind `author` und `editor`.

Wörterlisten werden analysiert und in die einzelnen Positionen und das Bindewort `and` aufgespalten, aber nicht weiter zerlegt. Wörtliche Listen können in der `bib`-Datei mit dem Stichwort `,and others` gekürzt werden. Es gibt zwei Unterarten:

Wörterlisten im engeren Sinne werden wie oben beschrieben behandelt. Die einzelnen Begriffe werden einfach ausgegeben. Typische Beispiele für solche wörtliche Listen sind `publisher` und `location`.

Schlüssellisten sind eine Variante der Wörterlisten, die druckfähige Daten oder Lokalisierungsschlüssel enthalten können. Für jedes Element in der Liste, wird ein Test durchgeführt, um festzustellen, ob es ein bekannter Lokalisationsschlüssel ist (die standardmäßigen Lokalisationsschlüssel sind in § 4.9.2 aufgeführt). Wenn ja, wird die lokalisierte Zeichenfolge dargestellt. Wenn nicht, wird das Element ausgegeben. Ein typisches Beispiel für eine Schlüsselliste ist `language`.

Felder werden in der Regel als Ganzes ausgegeben. Es gibt mehrere Subtypen:

Literal fields Wortwörtliche Felder werden ausgegeben, wie sie sind. Typische Beispiele für Wortfelder sind `title` und `note`.

Range fields sind Wortfelder mit einer Besonderheit: Alle Bindestriche sind normiert und durch den Befehl `\bibrangedash` ersetzt. Eine beliebige Anzahl von aufeinander folgenden Bindestrichen ergibt nun nur einen einzigen Bindestrich. Ein typisches Beispiel für einen Feldbereich ist das Feld `pages`.

Integer fields beinhalten unformatiert Zahlen, die in Ordnungszahlen oder Zeichenfolgen umgewandelt werden können, falls sie ausgegeben werden. Ein typisches Beispiel ist das Feld `month`.

Date fields beinhalten eine Datumsangabe im `yyyy-mm-dd` Format oder einen Datumsbereich in `yyyy-mm-dd/yyyy-mm-dd` Format. Datumsfelder sind in der Hinsicht speziell, da sie analysiert und in ihre Komponenten aufgespalten werden. Siehe § 2.3.8 für Details. Ein typisches Beispiel ist `date`.

Verbatim fields sind im wörtlichen Modus und können Sonderzeichen enthalten. Typische Beispiele für solche Felder sind `url` und `doi`.

Key fields Schlüsselfelder enthalten druckbare Daten oder Lokalisierungsschlüssel. Es wird ein Test durchgeführt, um festzustellen, ob der Wert des Feldes ein bekannter Lokalisierungsschlüssel ist (die standardmäßig definierten Lokalisierungsschlüssel werden in § 4.9.2 aufgeführt). Wenn ja, wird die lokalisierte Zeichenfolge gedruckt. Wenn nicht, wird der Wert ausgegeben. Ein typisches Beispiel ist das Feld `type`.

Special fields Spezialfelder sind Felder, die ein spezielles, oben noch nicht erwähntes, Format benötigen. Die Feldbeschreibung enthält in diesem Fall Details über das gewünschte Format. Typische Beispiele sind die Felder `hyphenation`, `keywords`, und `gender` aus § 2.2.3.

Der Datentyp eines Feldes wird in der Regel nicht geändert, unabhängig davon welcher Eingabetyp genutzt wird. Es gibt jedoch ein paar Sonderfälle. Die Feldbeschreibungen in § 2.2.2 beinhalten die Details oder Verweise der entsprechenden

Abschnitte in diesem Handbuch. Zum Beispiel das Feld `location`, welches standardmäßig eine Wörterliste ist, wird zur Schlüsselliste, wenn es in `@patent`-Einträgen verwendet wird. Das `series`-Feld, welches standardmäßig ein Wortfeld ist, wird bei Nutzung von `@article` Einträgen gesondert behandelt, siehe § 2.3.7 für Einzelheiten.

2.2.2. Datenfelder

Die Felder in diesem Abschnitt sind diejenigen, die regulär druckfähige Daten enthalten. Der Name auf der linken Seite ist der Name des Feldes, der von BibTeX und biblatex verwendet wird. Der biblatex Datentyp ist rechts neben dem Name angegeben. Siehe § 2.2.1 für Erläuterungen zu der verschiedenen Datentypen.

`abstract` field (literal)

Dieses Feld ist für die Aufnahme von Zusammenfassungen in eine `bib`-Datei, welche durch einen speziellen Bibliographiestil ausgegeben werden. Es wird nicht von allen Standard-Bibliographie Designs verwendet.

`addendum` field (literal)

Verschiedene bibliographische Daten, die am Ende des Eintrags gedruckt werden. Dies ist ähnlich dem `note`-Feld, außer dass es am Ende des Bibliographieeintrages gedruckt wird.

`afterword` list (Name)

Der/Die Autor/en von einem Nachwort zu einer Arbeit. Wenn der Autor des Nachwort mit dem `editor` und/oder `translator` übereinstimmt, verketteten die Standard-Stile diese Felder in der Bibliographie automatisch. Siehe auch `introduction` und `foreword`.

`annotation` field (literal)

Dieses Feld kann nützlich sein bei der Umsetzung eines Stils für kommentierte Bibliographien. Es wird nicht von allen Standard-Bibliographie Designs verwendet. Beachten Sie, dass dieses Feld nichts mit `annotator` zu tun hat. Der `annotator` ist der Autor von Anmerkungen, welche Teile der Arbeit zitiert.

`annotator` list (Name)

Der/Die Autor/en der Anmerkungen zum Werk. Wenn der Kommentator mit dem `editor` und/oder `translator` übereinstimmt, verketteten die Standard-Stile diese Felder in der Bibliographie automatisch. Siehe auch `commentator`.

- author** list (Name)
Der/Die Autor/en des `title`.
- author**type field (Schlüssel)
Die Art des Autors. Dieses Feld wirkt sich auf die Zeichenfolge (sofern vorhanden) aus, um den Autor einzuführen. Wird nicht von den Standard Bibliographie Designs verwendet.
- bookauthor** list (Name)
Der/Die Autor/en des `booktitle`.
- bookpagination** field (Schlüssel)
Wenn die Arbeit als Teil einer anderen veröffentlicht wird, ist dies das Seitennummerierungsschemas der Arbeit, d. h. `bookpagination` gehört zu `pagination` wie `booktitle` zu `title`. Der Wert dieses Feldes beeinflusst die Formatierung der Felder `pages` und `pagetotal`. Der Schlüssel sollte im Singular angegeben werden. Mögliche Schlüssel sind `page`, `column`, `line`, `verse`, `section`, und `paragraph`. Siehe auch `pagination` sowie § 2.3.10.
- booksubtitle** field (literal)
Der Untertitel, der zu dem `booktitle` gehört. Wenn das `subtitle`-Feld auf eine Arbeit verweist, die Teil einer größeren Veröffentlichung ist, ist ein möglicher Untertitel der Hauptarbeit in diesem Feld anzugeben. Siehe auch `subtitle`.
- booktitle** field (literal)
Wenn das Feld `title` den Titel einer Arbeit enthält, die Teil einer größeren Publikation ist, wird der Titel des Hauptwerkes in diesem Feld angegeben. Siehe auch `title`.
- booktitleaddon** field (literal)
Ein Anhang zum `booktitle`, der in einer anderen Schrift dargestellt wird.
- chapter** field (literal)
Ein Kapitel oder Abschnitt oder eine andere Einheit eines Werkes.
- commentator** list (Name)
Der/Die Autor/en eines Kommentars über die Arbeit. Beachte, dass dieses Feld für kommentierte Ausgaben, die einen Kommentator zusätzlich zu dem Autor haben,

bestimmt ist. Wenn die Arbeit ein eigenständiger Kommentar ist, sollte der Kommentator im Feld `author` angegeben werden. Wenn der Kommentator mit dem `editor` und/oder `translator` übereinstimmt, verketteten die Standard-Stile diese Felder in der Bibliographie automatisch. Siehe auch `annotator`.

- `date` field (Datum)
Zeitpunkt der Publikation. Siehe auch `month` und `year`, sowie § 2.3.8.
- `doi` field (verbatim)
Der digitale Objektbezeichner des Werkes.
- `edition` field (Integer oder literal)
Die Ausgabe einer gedruckten Publikation. Dies muss eine ganze Zahl und nicht eine Ordnungszahl sein. Nicht `edition={First}` oder `edition={1st}`, sondern `edition={1}`. Die Bibliographie-Stile wandeln diese in Ordnungssprachen um. Es ist auch möglich, die Ausgabe als Zeichenkette anzugeben, zum Beispiel „Dritte, überarbeitete und erweiterte Auflage“.
- `editor` list (Name)
Der/Die Editor/en von `title`, `booktitle` oder `maintitle`, je nach Eingabetyp Typ. Verwenden Sie das Feld `editortype`, um anzugeben, wenn es verschieden von ‚`editor`‘ ist. Siehe § 2.3.6 für weitere Hinweise.
- `editora` list (Name)
Ein zweiter Herausgeber, der eine andere redaktionelle Rolle innehat, wie die Erstellung, Schwärzen, etc. Nutzen Sie das `editoratype` Feld, um dies zu spezifizieren. Siehe § 2.3.6 für weitere Hinweise.
- `editorb` list (Name)
Ein weiterer sekundärer Herausgeber, der eine weitere redaktionelle Rolle innehat. Verwenden Sie das Feld `editorbtype`, um dies anzugeben. Siehe § 2.3.6 für weitere Hinweise.
- `editorc` list (Name)
Ein weiterer sekundärer Herausgeber, der eine weitere redaktionelle Rolle innehat. Verwenden Sie das Feld `editorctype`, um dies anzugeben. Siehe § 2.3.6 für weitere Hinweise.

editortype field (Schlüssel)

Die Art der redaktionellen Rolle, die der editor hat. Rollen, die standardmäßig unterstützt werden, sind `editor`, `compiler`, `founder`, `continuator`, `redactor`, `collaborator`. Die Rolle `editor` ist die Standardeinstellung. In diesem Fall kann das Feld weggelassen werden. Siehe § 2.3.6 für weitere Hinweise.

editoratype field (Schlüssel)

Ähnlich wie `editortype`, aber auf `editora` bezogen. Siehe § 2.3.6 für weitere Hinweise.

editorbtype field (Schlüssel)

Ähnlich wie `editortype`, aber auf `editorb` bezogen. Siehe § 2.3.6 für weitere Hinweise.

editorctype field (Schlüssel)

Ähnlich wie `editortype`, aber die `editorc` bezogen. Siehe § 2.3.6 für weitere Hinweise.

eid field (literal)

Die elektronische Kennzeichnung eines `@article`.

eprint field (literal)

Die elektronische Kennzeichnung einer Online-Publikation. Dies ist ungefähr vergleichbar mit einem `doi`, aber spezifisch für ein bestimmtes Archiv, Quelle, Service oder System. Siehe § 3.10.6 für Einzelheiten. Siehe auch `eprinttype` und `eprintclass`.

eprintclass field (literal)

Weitere Informationen, die durch `eprinttype` gegeben sind. Dies könnten ein Teil eines Archivs, ein Dateipfad, oder eine Klassifizierung etc. sein. Siehe § 3.10.6 für Details. Siehe auch `eprint` und `eprinttype`.

eprinttype field (literal)

Die Art der `eprint` Kennung, d. h., der Name des Archivs, Quelle, Service, oder des System, zu dem `eprint` gehört. Siehe § 3.10.6 für Details. Siehe auch `eprint` und `eprintclass`.

eventdate field (Datum)

Das Datum der Konferenz, eines Symposiums oder einer anderen Veranstaltung, die in `@proceedings` und `@inproceedings` eingetragen ist. Dieses Feld kann auch für die benutzerdefinierte Typen, die in § 2.1.3 aufgelistet sind, nützlich sein. Siehe auch `eventtitle` und `venue`, sowie auch § 2.3.8.

eventtitle field (literal)

Der Titel der Konferenz, eines Symposiums oder einer anderen Veranstaltung, die in `@proceedings` und `@inproceedings` eingetragen ist. Dieses Feld kann auch für die benutzerdefinierte Typen, die in § 2.1.3 aufgelistet sind, nützlich sein. Beachten Sie, dass dieses Feld den schlichten/nüchteren Titel der Veranstaltung enthält. Dinge wie „Verfahren der Fünften XYZ-Konferenz“ werden in `titleaddon` oder `booktitleaddon` eingetragen. Siehe auch `eventdate` und `venue`.

file field (literal)

Ein lokaler Link zu einer pdf- oder anderen Version des Werkes. Wird nicht von den Standardbibliographien verwendet.

foreword list (Name)

Der/Die Autor/en eines Vorwortes zum Werk. Wenn der Autor des Vorwortes mit dem `editor` und/oder `translator` übereinstimmt, verketteten die Standardstile diese Felder in der Bibliographie automatisch. Siehe auch `introduction` und `afterword`.

holder list (Name)

Der/Die Inhaber eines `@patent`, falls verschieden von `author`. Die kooperierenden Inhaber müssen nicht zusätzlich in Klammern gesetzt werden, siehe § 2.3.3 für weitere Einzelheiten. Diese Liste kann auch für die benutzerdefinierte Typen aus § 2.1.3 nützlich sein.

howpublished field (literal)

Eine veröffentlichte Bekanntmachung für eine außergewöhnliche Publikation, die in keine der üblichen Kategorie passt.

indextitle field (literal)

Ein Titel für eine Indizierung anstelle des regulären `title`-Feldes. Dieses Feld kann nützlich sein, wenn Sie möchten, das ein Eintrag wie „Eine Einführung in die ...“ als „Einführung in die ..., Eine“ indiziert wird. Stilautoren sollten beachten, dass `biblatex` automatisch den Wert von `title` in `indextitle` kopiert, wenn letzteres undefiniert ist.

institution list (literal)

Der Name der Universität oder einer anderen Institution, je nach Art des Eintrags. Herkömmlich benutzt BibTeX dafür das Feldes `school`, welches als Alias unterstützt wird. Siehe auch §§ 2.2.5 and 2.3.4.

introduction list (Name)

Der/Die Autor/en der Einführung in die Arbeit. Wenn der Autor der Einführung mit `editor` und/oder `translator` übereinstimmt, verketteten die Standardstile diese Felder in der Bibliographie automatisch. Siehe auch `foreword` und `afterword`.

isan field (literal)

Die International Standard Audiovisual Number eines audiovisuellen Werkes. Wird nicht von den Standardbibliographiestilen verwendet.

isbn field (literal)

Die International Standard Book Number eines Buches.

ismn field (literal)

Die International Standard Music Number für gedruckte Musik wie Notenbücher. Wird nicht von den Standardbibliographiestilen verwendet.

isrn field (literal)

Die International Standard Technical Report Number eines technischen Berichts.

issn field (literal)

Die International Standard Serial Number einer Zeitschrift.

issue field (literal)

Die Ausgabe einer Zeitschrift. Dieses Feld ist für Zeitschriften, deren einzelne Ausgaben durch Bezeichnung wie ‚Frühling‘ oder ‚Sommer‘ anstatt des Monats oder einer Nummer identifiziert werden. Da die Verwendung von `issue` ähnlich zu `month` und `number` ist, kann dieses Feld auch für Doppelausgaben und andere Sonderfälle sinnvoll sein. Siehe auch `month`, `number` und § 2.3.9.

issuesubtitle field (literal)

Der Untertitel einer bestimmten Ausgabe einer Zeitschrift oder einer anderen Zeitung.

issuetitle field (literal)

Der Titel einer bestimmten Ausgabe einer Zeitschrift oder einer anderen Zeitung.

iswc field (literal)

Der Internationale Standard Work Code eines musikalischen Werkes. Wird nicht von den Standardbibliographiestilen verwendet.

journalsubtitle field (literal)

Der Untertitel eines Journals, einer Zeitung oder einer anderen Zeitschrift.

journaltitle field (literal)

Der Name eines Journals, einer Zeitung oder einer anderen Zeitschrift.

label field (literal)

Eine Bezeichnung, die bei den Zitierstilen als Ersatz für standardmäßige Labels verwendet wird, wenn alle erforderlichen Daten zur Generierung des standardmäßigen Labels fehlen. Zum Beispiel, wenn eine Autor-Jahr-Zitierweise generiert wird, der Autor oder das Jahr fehlt, kann es in `label` angegeben werden. Siehe § 2.3.2 für weitere Einzelheiten. Beachten Sie, dass im Gegensatz zu `shorthand`, `label` als Absicherung verwendet wird. Siehe auch `shorthand`.

language list (key)

Die Sprache/n des Werkes. Sprachen können wörtlich oder als Lokalisierungsschlüssel angegeben werden. Wenn Lokalisierungsschlüssel verwendet werden, ist das Präfix `lang` wegzulassen. Siehe auch `origlanguage` und `hyphenation` in § 2.2.3.

library field (literal)

Dieses Feld kann nützlich sein, um Informationen über beispielsweise den Namen einer Bibliothek und eine Signatur anzugeben. Dies kann durch einen speziellen Bibliographiestil erreicht werden, falls gewünscht. Wird nicht von den Standardbibliographiestilen verwendet.

location list (literal)

Der/Die Ort/e der Veröffentlichung, d. h. der Standort des `publisher` oder der `institution`, je nach Art des Eingabetyps. Herkömmliche benutzt BibTeX das Feld `address`, welches ähnlich genutzt wird. Siehe auch §§ 2.2.5 and 2.3.4. Mit `@patent`-Einträgen zeigt diese Liste den Umfang eines Patents und wird als ein Schlüsselliste behandelt. Diese Liste kann auch für die benutzerdefinierte Typen aus § 2.1.3 nützlich sein.

mainsubtitle field (literal)

Der Untertitel der zum maintitle gehört. Siehe auch subtitle.

maintitle field (literal)

Die Haupttitel eines mehrbändigen Buches, wie bei *Gesammelten Werken*. Wenn title oder booktitle den Titel eines einzigen Bandes, welches Teil eines mehrbändigen Buches ist, ist der Titel der gesamten Arbeit in diesem Feld enthalten.

maintitleaddon field (literal)

Ein Anhang zu maintitle, der in einer anderen Schrift ausgegeben wird.

month field (integer)

Der Monat der Veröffentlichung. Dies muss eine ganze Zahl und nicht eine Ordnungszahl oder eine Zeichenkette sein. Nicht month={January}, sondern month={1}. Die Bibliographie-Stile wandelt dies, falls erforderlich, in eine Zeichenkette oder Ordnungszahl, der jeweiligen Sprache entsprechend, um. Siehe auch date sowie §§ 2.3.9 and 2.3.8.

nameaddon field (literal)

Ein Zusatz direkt hinter dem Namen des Autors, der in der Bibliographie angegeben wird. Wird nicht von den Standardbibliographiestilen verwendet. Dieses Feld kann nützlich sein, um ein Aliasnamen oder ein Pseudonym hinzuzufügen (oder der richtige Namen, wenn das Pseudonym genutzt wird, um sich auf den Verfasser zu beziehen).

note field (literal)

Verschiedene bibliographische Daten, die nicht in ein anderes Feld passen. note wird verwendet, um bibliographische Daten in einem freien Format aufzuzeichnen. Fakten über die Veröffentlichung, wie „Nachdruck der Ausgabe London 1831“ sind typische Kandidaten für dieses Feldes. Siehe auch addendum.

number field (literal)

Die Nummer einer Zeitschrift oder die Ausgabe/Nummer eines Buches aus einer series. Siehe auch issue sowie §§ 2.3.7 and 2.3.9. Mit @patent ist dies die Anzahl oder Eintragsnummer eines Patents oder von Patentrechteanfragen.

organization list (literal)

Die Organisation, die ein @manual oder eine @online-Quelle veröffentlichte oder eine Konferenz sponserte. Siehe auch § 2.3.4.

origdate field (date)

Wenn das Werk eine Übersetzung, eine Neuauflage oder etwas Ähnliches ist, dann das Datum der Veröffentlichung der Originalausgabe. Wird nicht von den Standardbibliographiestilen verwendet. Siehe auch `date`.

origlanguage field (key)

Wenn das Werk eine Übersetzung ist, dann die Sprache des Originals. Siehe auch `language`.

origlocation list (literal)

Wenn das Werk eine Übersetzung, eine Neuauflage oder etwas Ähnliches ist, dann `location` der Originalausgabe. Wird nicht von den Standardbibliographiestilen verwendet. Siehe auch `location` und § 2.3.4.

origpublisher list (literal)

Wenn das Werk eine Übersetzung, eine Neuauflage oder etwas Ähnliches ist, der `publisher` der Originalausgabe. Wird nicht von den Standardbibliographiestilen verwendet. Siehe auch `publisher` und § 2.3.4.

origtitle field (literal)

Wenn das Werk eine Übersetzung ist, der `title` des Originals. Wird nicht von den Standardbibliographiestilen verwendet. Siehe auch `title`.

pages field (Bereich)

Eine oder mehrere Seitenzahlen oder Seitenbereiche. Wenn das Werk als Teil eines anderen veröffentlicht wurde, wie ein Artikel in einer Zeitschrift oder einer Sammlung, enthält dieses Feld den entsprechenden Seitenbereich der anderen Arbeit. Es kann auch verwendet werden, um den Bezug auf einen bestimmten Teil eines Werkes zu begrenzen (beispielsweise ein Kapitel in einem Buch).

pagetotal field (literal)

Die Gesamtzahl der Seiten des Werkes.

pagination field (key)

Die Seitennummerierung der Arbeit. Der Wert dieses Feldes legt die Formatierung des `<postnote>`-Argumentes für einen Zitatbefehl fest. Der Schlüssel sollte im Singular angegeben werden. Mögliche Schlüssel können sein: `page`, `column`, `line`, `verse`, `section` und `paragraph`. Siehe auch `bookpagination` sowie §§ 2.3.10 und 3.11.3.

part field (literal)

Die Nummer einer Teilausgabe. Dieses Feld gilt nur für Bücher und nicht für Fachzeitschriften. Es kann verwendet werden, wenn ein logischer Band aus zwei oder mehreren physischen besteht. In diesem Fall wird die Nummer des logischen Bandes in `volume` geschrieben und die Nummer des Teils in `part`. Siehe auch `volume`.

publisher list (literal)

Der/Die Name/n des/der Herausgeber. Siehe auch § 2.3.4.

pubstate field (key)

Der Stand der Veröffentlichung der Arbeiten, z. B. ‚im Druck‘ oder ‚eingereicht‘ (bei einer Zeitschrift). Bekannte Stadien sind `inpress` und `submitted`.

reprinttitle field (literal)

Der Titel einer Neuauflage des Werkes. Wird nicht von den Standardstilen verwendet.

series field (literal)

Der Name einer Publikationsreihe, wie „Studien in ...“, oder die Nummer einer Zeitschriftenreihe. Bücher in einer Publikationsreihe sind in der Regel nummeriert. Die Nummer oder Ausgabe eines Buches in einer Serie, wird in `number` eingetragen. Beachten Sie, dass auch `@article` die Eingabe von `series` nutzt, aber es in besonderer Weise handhaben. Siehe § 2.3.7 für weitere Einzelheiten.

shortauthor list (name)

Der/Die Autor/en der Arbeit in abgekürzter Form. Dieses Feld soll vor allem für Abkürzungen von kooperierenden Autoren genutzt werden, siehe § 2.3.3 für weitere Einzelheiten.

shorteditor list (name)

Der/Die Herausgeber des Werkes, in abgekürzter Form. Dieses Feld soll vor allem zur Abkürzung von kooperierenden Herausgebern genutzt werden, siehe § 2.3.3 für weitere Einzelheiten.

shorthand field (literal)

Eine besondere Bezeichnung, die bei der Zitierweise anstelle der üblichen verwendet wird. Dieses Feld ist für die Aliaszitierung vorgesehen. Wenn definiert, überschreibt es das standardmäßige Label. Wenn einer der zitierten Bibliographieinträge ein `shorthand`-Feld enthält, erstellt `biblatex` automatisch eine Liste von Kürzeln, die

zusätzlich zu der normalen Bibliographie ausgegeben wird. Siehe § 3.5.3 für weitere Einzelheiten. Siehe auch `label`.

`shorthandintro` field (literal)

Der ausführliche Zitierstil, der mit diesem Paket geliefert wird, benutzt Sätze, wie „fortan zitiert als [Kurzform]“, um Abkürzungen im ersten Zitat einzuführen. Wenn der `shorthandintro`-Bereich definiert ist, überschreibt es den Standardsatz. Beachten Sie, dass die alternativen Sätze Abkürzungen enthalten müssen.

`shortjournal` field (literal)

Eine kurze Version oder eine Abkürzung von `journaltitle`. Wird nicht von den Standardbibliographiestilen verwendet.

`shortseries` field (literal)

Eine kurze Version oder eine Abkürzung von `series`. Wird nicht von den Standardbibliographiestilen verwendet.

`shorttitle` field (literal)

Der Titel in einer verkürzten Form. Dieses Feld ist in der Regel nicht in der Bibliographie enthalten. Es ist für Zitierungen im Autor-Titel-Format vorgesehen. Falls vorhanden, verwendet der Autor-Titel-Zitatstil dieses Feld anstatt `title`.

`subtitle` field (literal)

Der Untertitel des Werkes.

`title` field (literal)

Der Titel des Werkes.

`titleaddon` field (literal)

Ein Anhang zum `title`, der in einer anderen Schrift ausgegeben wird.

`translator` list (name)

Der/Die Übersetzer von `title` oder `booktitle`, je nach Art des Eintrags. Wenn der Übersetzer mit dem `editor` übereinstimmt, verketteten die Standardstile diese Felder in der Bibliographie automatisch.

`type` field (key)

Die Art von `manual`, `patent`, `report`, oder `thesis`. Dieses Feld kann auch für die benutzerdefinierte Typen aus § 2.1.3 nützlich sein.

`url` field (literal)

Die URL einer Online-Publikation.

`urldate` field (Datum)

Der Zugriffszeitpunkt auf die Adresse, die im `url`-Feld angegeben ist. Siehe auch § 2.3.8.

`venue` field (literal)

Der Veranstaltungsort einer Konferenz, eines Symposiums oder eine andere Veranstaltung in `@proceedings`- und `@inproceedings`-Einträgen. Dieses Feld kann auch für die benutzerdefinierte Typen aus § 2.1.3 nützlich sein. Beachten Sie, dass `location` den Ort der Veröffentlichung enthält. Es entspricht daher der `publisher`- und `institution`-Liste. Der Veranstaltungsort der Veranstaltung wird in `venue` eingetragen. Siehe auch `eventdate` und `eventtitle`.

`version` field (literal)

Die Revisionsnummer einer Software, eines Handbuches, etc.

`volume` field (literal)

Das Nummer eines mehrbändigen Buches oder eine Zeitschrift. Siehe auch `part`.

`volumes` field (literal)

Die Gesamtzahl der Bände eines mehrbändigen Werkes. Je nach Art des Eintrags gehört dieses Feld zu `title` oder `maintitle`.

`year` field (literal)

Das Jahr der Veröffentlichung. Siehe auch `date` und § 2.3.8.

2.2.3. Spezielle Felder

Die Felder in diesem Abschnitt werden nicht ausgegeben, sondern dienen einem anderen Zweck. Sie gelten für alle Arten von Einträgen.

`crossref` field (entry key / Eingabeschlüssel)

Dieses Feld enthält einen Eingabeschlüssel für die BibTeX = "Querverweisfunktion. Es wird intern von BibTeX verwendet. Abhängige „Kind“einträge mit einem `crossref`-Feld erben alle Daten von den Elterneinträgen, die im `crossref`-Feld angegeben sind. Wenn die Anzahl der Kindeinträge auf einen bestimmten Elterneintrag verweist, wird der Elterneintrag automatisch in die Bibliographie hinzugefügt, auch

wenn er nicht explizit zitiert wurde. Man kann dies über die Kommandozeile steuern.¹¹ Stilautoren sollten beachten, dass das `crossref`-Feld des Kindeintrags auf der `biblatex`-Ebene definiert wird, abhängig von der Verfügbarkeit der Elterneinträge. Wenn der Elterneintrag verfügbar ist, wird das `crossref`-Feld des Kindeintrages definiert. Wenn nicht, erben sie die Daten aus den Elterneinträgen, aber ihr `crossref` Feld bleibt undefiniert. Ob der Elterneintrag implizit oder explizit in die Bibliographie aufgenommen wurde, spielt keine Rolle. Siehe auch `xref`.

entryset field (comma-separated values / durch Kommata getrennter Wert)

Dieses Feld ist speziell für `@set`-Elterneinträge und deren Kindereinträge. Die Elterneinträge sind eine durch Kommas getrennte Liste der Eingabeschlüssel, die einen Eintrag festlegen. In den Kindeinträgen ist es der Eingabeschlüssel der Eltern. Siehe § 3.10.5 für Details.

entrysubtype field (identifier / Bezeichner)

Dieses Feld, welches nicht durch die Standardstile unterstützt wird, kann verwendet werden, um einen Subtyp eines Eingabetyps anzugeben. Dies kann für Bibliographiestile nützlich sein, die eine feiner Abstufung von Eingabetypen unterstützen.

execute field (code)

Ein spezielles Feld, welches beliebige TeX-Codes enthält, die ausgeführt werden können, wenn auf die Daten von den entsprechenden Einträgen zugegriffen wird. Dies kann nützlich sein, um spezielle Fälle zu behandeln. Dieses Feld ist vergleichbar mit den Schnittstellen `\AtEveryBibitem`, `\AtEveryLositem` und `\AtEveryCitekey` aus § 4.10.6, außer dass es auf einer pro-Eintrag Grundlage in der `bib`-Datei definierbar ist. Jeder Code dieses Felds wird direkt automatisch nach den Hookees ausgeführt.

gender field (`sf`, `sm`, `sn`, `pf`, `pm`, `pn`, `pp`)

Das Geschlecht des Autors oder das Geschlecht des Herausgebers, wenn es keinen Autor gibt. Die folgenden Kennungen werden unterstützt: `sf` (femininer Singular, ein einzelner weiblicher Name), `sm` (maskuliner Singular, ein einzelner männlicher Name), `sn` (Neutrum Singular, ein einzelner neutraler Name), `pf` (femininer Plural, mehrere weibliche Namen), `pm` (maskuliner Plural, mehrere männliche Namen), `pn` (Neutrum Plural, mehrere neutrale Namen), `pp` (Plural, mehrere Namen unterschiedlichen Geschlechts). Diese Information ist nur für spezielle Bibliographie- und Zitierstile erforderlich und auch nur in bestimmten Sprachen. Zum Beispiel kann ein Zitatstil sich wiederholender Autorennamen mit einem Begriff wie ‚idem‘ ersetzen. Wenn das lateinische Wort verwendet wird, so wie es im englischen und französischen Brauch ist, muss das Geschlecht nicht angegeben werden. In deutschen

¹¹siehe die `mincrossrefs` package option in § 3.1.2.1.

Sprache	Region/Dialekt	Babel Identifiers
Dänisch	Dänemark	danish
Niederländisch	Niederlande	dutch
Englisch	USA	american, USenglish, english
	Großbritannien	british, UKenglish
	Kanada	canadian
	Australien	australian
	Neuseeland	newzealand
Finnisch	Finnland	finnish
Französisch	Frankreich, Kanada	french, francais, canadien
Deutsch	Deutschland	german, ngerman
	Österreich	austrian, naustrian
Griechisch	Griechenland	greek
Italienisch	Italien	italian
Norwegisch	Norwegen	norsk, nynorsk
Portugiesisch	Brasilien	brazil
	Portugal	portuges
Spanisch	Spanien	spanish
Schwedisch	Schweden	swedish

Tabelle 1: Unterstützte Sprachen

Veröffentlichungen werden jedoch solche Schlüsselbegriffe in der Regel in Deutsch gegeben und sind in diesem Fall geschlechtsspezifisch.

hyphenation field (identifier / Bezeichner)

Die Sprache der Bibliographieinträge. Der Bezeichner muss eine Sprache sein, die dem Paket `babel` bekannt ist. Diese Informationen können verwendet werden, um zwischen Silbentrennungsmustern und Zeichenketten in der Bibliographie zu wechseln. Beachten Sie die Groß- und Kleinschreibung bei Sprachen. Die Sprachen, die derzeit von diesem Paket unterstützt werden, sind in Tabelle 1 zusammengefasst. Beachten Sie, dass `babel` den Bezeichner `english` ähnlich wie `british` oder `american` behandelt, abhängig von der Version. Das Paket `biblatex` behandelt es immer ähnlich wie `american`. Es ist besser, die Sprachbezeichner `american` und `british` zu verwenden, um möglichen Missverständnissen vorzubeugen. Vergleichen Sie `language` in § 2.2.2.

indexsorttitle field (literal)

Der Titel beim Sortieren des Indexes. Im Gegensatz zu `indextitle` wird dieses Feld nur zum Sortieren verwendet. Der ausgegebene Titel im Index ist `indextitle` oder `title`. Dieses Feld kann nützlich sein, wenn der Titel Sonderzeichen oder Befehle

enthält, die die Sortierung des Indexes beeinträchtigen. Betrachten Sie folgendes Beispiel:

```
title          = {The \LaTeX\ Companion},
indextitle     = {\LaTeX\ Companion, The},
indexsorttitle = {LATEX Companion},
```

Stilautoren sollten beachten, dass `biblatex` automatisch den Wert dieser beiden Felder `indextitle` oder `title` in `indexsorttitle` kopiert, falls dieses Feld nicht definiert ist.

keywords field (comma-separated values / durch Kommata getrennter Wert)

Eine durch Kommata getrennte Liste von Schlüsselwörtern. Diese Schlüsselwörter sind für die Bibliographiefilter (vgl. §§ 3.5.2 and 3.10.4), welche normalerweise nicht ausgegeben werden. Beachten Sie, dass Leerzeichen nach Kommata ignoriert werden.

options field (comma-separated $\langle key \rangle = \langle value \rangle$ Option)

Eine kommaseparierte Liste von Eingabemöglichkeiten in $\langle key \rangle = \langle value \rangle$ -Notation. Dieses Feld wird verwendet, um Optionen auf einer pro-Eintrag Grundlage festzulegen. Siehe § 3.1.3 für weitere Einzelheiten. Beachten Sie, dass Zitier- und Bibliographiestile zusätzliche Eingabemöglichkeiten haben können.

presort field (string / Zeichenkette)

Ein besonderes Feld, um die Sortierung der Bibliographie zu modifizieren. Dieses Feld ist das erste Element der Sortierung, das beim Sortieren der Bibliographie verwendet wird und damit wird es verwendet, um die Einträge in Gruppen zu unterteilen. Dies kann bei der Erstellung unterteilter Bibliographien mit den Bibliographiefiltern nützlich sein. Dieses Feld wird nur intern bei der Sortierung verwendet. Bitte beachten Sie § 3.4 für weitere Details. Siehe auch § 4.5.1.

sortkey field (literal)

Ein Feld, um die Sortierreihenfolge der Bibliographie zu ändern. Benutzen Sie dieses Feld als den Hauptsortierschlüssel. Falls vorhanden, nutzt `biblatex` dieses Feld während der Sortierung und ignoriert alles andere, mit Ausnahme des `presort`-Feldes. Dieses Feld wird nur intern während des Sortierens verwendet. Bitte beachten Sie § 3.4 für weitere Details.

sortname list (name)

Ein Name oder eine Liste von Namen, um die Sortierreihenfolge der Bibliographie zu verändern. Wenn diese Liste vorhanden ist, wird sie anstelle von `author` oder

`editor` genutzt, um die Bibliographie zu sortieren. Es wird nur intern während des Sortierens verwendet. Bitte beachten Sie § 3.4 für weitere Details.

`sortshorthand` field (literal)

Biber only

Ähnlich wie `sortkey`, wird aber in der Abkürzungsliste verwendet. Wenn vorhanden, verwendet `biblatex` dieses Feld anstelle von `shorthand` zur Abkürzungssortierung. Dies ist nützlich, wenn das Abkürzungsfeld Formatierungsbefehle wie `\emph` oder `\textbf` enthält. Dieses Feld wird nur bei der internen Sortierung eingesetzt.

`sorttitle` field (literal)

Ein Feld, um die Sortierreihenfolge der Bibliographie zu modifizieren. Falls vorhanden, wird dieses Feld anstelle von `title` beim Sortieren der Bibliographie verwendet. Es wird nur intern während der Sortierung verwendet. Das `sorttitle`-Feld kann nützlich sein, wenn Sie einen Eintrag mit einem Titel wie „An Introduction to...“ haben, und ihn alphabetisch unter ‚I‘ anstatt ‚A‘ haben wollen. In diesem Fall können Sie „Introduction to...“ in `sorttitle` eintragen. Bitte beachten Sie § 3.4 für weitere Details.

`sortyear` field (literal)

Ein Feld, um die Sortierreihenfolge der Bibliographie zu modifizieren. Falls vorhanden, wird dieses Feld anstelle von `year` beim Sortieren der Bibliographie verwendet. Es wird nur intern während der Sortierung verwendet. Bitte beachten Sie § 3.4 für weitere Details.

`xref` field (key)

Dieses Feld bietet einen alternativen Querverweismechanismus. Es unterscheidet sich von `crossref` insofern, dass der Kindeintrag nicht alle Daten von den Elterneinträgen erbt, die im `xref`-Feld angegeben sind. Wenn die Anzahl der Kindeinträge auf einen bestimmten Elterneintrag verweist, wird dieser automatisch der Bibliographie hinzugefügt, auch wenn er nicht explizit zitiert wurde. Diese Schwelle ist mit der Option des `mincrossrefs` Paketes aus § 3.1.2.1 einstellbar. Stilautoren sollten beachten, dass `xref`-Felder der Kindeinträge auf der `biblatex` definiert werden, abhängig von der Verfügbarkeit der Elterneinträge. Wenn die Elterneinträge vorhanden sind, werden die `xref`-Felder der Kindeinträge definiert. Wenn nicht, bleibt `xref` undefiniert. Ob die Elterneinträge in die Bibliographie implizit oder explizit zugefügt wurden, spielt keine Rolle. Bitte beachten Sie § 2.4.1 für weitere Erläuterungen. Siehe auch `crossref`.

2.2.4. Benutzerdefinierte Felder

Die Felder in diesem Abschnitt sind für spezielle Bibliographiestile bestimmt. Sie werden nicht in den Standardbibliographiestilen verwendet.

`name[a-c]` list (Name)

Benutzerdefinierte Listen für spezielle Bibliographiestile. Wird nicht von den Standardbibliographiestilen verwendet.

`name[a-c]type` field (Schlüssel)

Ähnlich wie `authortype` und `editortype`, aber bezogen auf `name[a-c]`. Wird nicht von den Standardbibliographiestilen verwendet.

`list[a-f]` list (literal)

Benutzerdefinierte Listen für spezielle Bibliographiestile. Wird nicht von den Standardbibliographiestilen verwendet.

`user[a-f]` field (literal)

Benutzerdefinierte Felder für spezielle Bibliographiestile. Wird nicht von den Standardbibliographiestilen verwendet.

`verb[a-c]` field (literal)

Ähnlich wie die benutzerdefinierten Felder oben, außer dass es sich um wörtlich Felder handelt. Wird nicht von den Standardbibliographiestilen verwendet.

2.2.5. Aliasfeldnamen

Die Aliase in diesem Abschnitt sind für die Abwärtskompatibilität mit herkömmlichen BibTeX und anderen Anwendungen, die darauf basieren. Beachten Sie, dass diese Aliase sofort aufgelöst werden, sobald die `bib`-Datei verarbeitet wird. Alle Bibliographie- und Zitierstile müssen die Namen der Felder nutzen, auf die sie verweisen und nicht den Alias. In `bib`-Dateien können Sie entweder den Alias oder den Namen des Feldes benutzen, jedoch nicht beide gleichzeitig.

`address` list (literal)

Ein Alias für `location`, für BibTeX-Kompatibilität bereitgestellt. Herkömmlich verwendet BibTeX den leicht irreführenden Feldnamen `address` für den Ort der Veröffentlichung, d. h. den Ort des Verlages, während `biblatex` den generische Feldnamen `location` nutzt. Siehe §§ 2.2.2 and 2.3.4.

`annotate` field (literal)

Ein Alias für `annotation`, für `jurabib`-Kompatibilität bereitgestellt. Siehe § 2.2.2.

`archiveprefix` field (literal)

Ein Alias für `eprinttype` für arXiv-Kompatibilität bereitgestellt. Siehe §§ 2.2.2 und 3.10.6.

`journal` field (literal)

Ein Alias für `journaltitle`, für BibTeX-Kompatibilität bereitgestellt. Siehe § 2.2.2.

`key` field (literal)

Ein Alias für `sortkey`, für BibTeX-Kompatibilität bereitgestellt. Siehe § 2.2.3.

`pdf` field (literal)

Ein Alias für `file`, für JabRef-Kompatibilität bereitgestellt. Siehe § 2.2.2.

`primaryclass` field (literal)

Ein Alias für `eprintclass` für arXiv-Kompatibilität bereitgestellt. Siehe §§ 2.2.2 und 3.10.6.

`school` list (literal)

Ein Alias für `institution`, für BibTeX-Kompatibilität bereitgestellt. `institution` wird herkömmlich von BibTeX für technische Berichte verwendet, während `school` die Institution einer Diplomarbeit enthält. Das Paket `biblatex` verwendet das generische Feld `institution` in beiden Fällen. Siehe §§ 2.2.2 und 2.3.4.

2.3. Verwendungshinweise

Die Eingabetypen und Felder, die durch dieses Paket unterstützt werden, sollten zum größten Teil intuitiv für jedermann nutzbar sein, der mit BibTeX vertraut ist. Doch abgesehen von den zusätzlichen Typen und Feldern, die von diesem Paket zu Verfügung gestellt werden, werden einige der alten Bekannten in einer Art und Weise gehandhabt, die einer Erklärung benötigt. Dieses Paket enthält einige Kompatibilitätskodes für `bib`-Dateien, die mit einem herkömmlichen BibTeX-Stil erzeugt wurden. Leider ist es nicht möglich, alle älteren Dateien automatisch in den Griff zu bekommen, da das `biblatex`-Datenmodell etwas anders als das herkömmliche BibTeX ist. Daher benötigen solche `bib`-Dateien höchstwahrscheinlich eine Bearbeitung, um richtig mit diesem Paket funktionieren zu können. Zusammenfassend sind die folgenden Elemente von den herkömmlichen BibTeX-Stilen verschieden:

- Der Eingabetyp `@inbook`. Siehe §§ 2.1.1 und 2.3.1 für weitere Einzelheiten.
- Die Felder `institution`, `organization`, und `publisher` sowie die Aliase `address` und `school`. Siehe §§ 2.2.2, 2.2.5, 2.3.4 für weitere Details.

- Der Umgang mit bestimmten Titelarten. Siehe § 2.3.5 für weitere Einzelheiten
- Das Feld `series`. Siehe §§ 2.2.2 and 2.3.7 für Einzelheiten.
- Die Felder `year` und `month`. Siehe §§ 2.2.2, 2.3.8, 2.3.9 für Einzelheiten.
- Das Feld `edition`. See § 2.2.2 für Einzelheiten.
- Das Feld `key`. Siehe § 2.3.2 für Einzelheiten.

Die Nutzer des `jurabib`-Paketes sollten beachten, dass `shortauthor` wie eine Namensliste von `biblatex` behandelt wird, siehe § 2.3.3 für weitere Einzelheiten.

2.3.1. Der Eingabetyp `@inbook`

Verwenden Sie `@inbook` nur für einen in sich geschlossenen Teil eines Buches mit eigenem Titel. Es bezieht sich auf `@book` genau wie `@incollection` auf `@collection`. Siehe § 2.3.5 für Beispiele. Wenn Sie auf ein Kapitel oder einen Abschnitt eines Buches verweisen möchten, nutzen Sie einfach `book` und fügen Sie ein `chapter`- und/oder `pages`-Feld hinzu. Ob eine Bibliographie überhaupt Verweise auf Kapitel oder Abschnitte enthalten sollte, ist umstritten, da ein Kapitel keine bibliographische Einheit darstellt.

2.3.2. Fehlende und weggelassene Daten

Die Felder, die als ‚erforderlich‘ in § 2.1.1 angegeben wurden, sind nicht unbedingt in allen Fällen erforderlich. Die mitgelieferten Bibliographiestile dieses Paketes können mit wenig auskommen, wie `title` ausreicht für die meisten Typen. Ein Buch, das anonym veröffentlicht wurde, eine Zeitschrift ohne einen expliziten Herausgeber oder ein Softwarehandbuch ohne einen expliziten Autor sollten kein Problem darstellen, so weit es die Bibliographie betrifft. Zitierstile können jedoch andere Anforderungen haben. Zum Beispiel erfordert ein Autor-Jahr-Zitat offensichtlich einen `author/editor` und ein `year`.

Sie können in der Regel mit dem `label`-Feld einen Ersatz für die fehlenden Daten für Zitierungen bereitstellen. Wie das `label`-Feld verwendet wird, hängt von dem Zitierstil ab. Der Autor-Jahr-Zitierstil, der in diesem Paket enthalten ist, nutzt das `label`-Feld als Absicherung, wenn entweder `author/editor` oder `year` fehlen. Auf der anderen Seite nutzen die numerischen Designs dies nicht bei allem, da das numerische Schema unabhängig von den verfügbaren Daten ist. Das Autor-Titel-Design ignoriert es weitgehend, da der bloße `title` in der Regel für eine einheitliche Zitierform ausreichend ist, ein Titel wird erwartungsgemäß in jedem Fall zur Verfügung stehen. Das `label`-Feld kann auch verwendet werden, um die nicht-numerischen Teile des automatisch generierten `labelalpha` Feldes (alphabetische Zitierform) zu überschreiben. Siehe § 4.2.4 für weitere Einzelheiten.

Beachten Sie, dass herkömmliche BibTeX-Stile ein Feld `key` für die Alphabetisierung besitzen, falls sowohl `author` und `editor` fehlen. Das Paket `biblatex` behandelt `key` ähnlich wie `sortkey`. Darüber hinaus bietet es eine sehr feine Sortierungskontrolle, vgl. §§ 2.2.3 and 3.4 für Details. Das Paket `natbib` verwendet `key` als ein Absicherungslabel für Zitate. Verwenden Sie stattdessen das `label`-Feld.

2.3.3. Kooperierende Autoren und Herausgeber

Kooperierende Autoren und Herausgeber werden jeweils in `author`- oder `editor`-Feld angeführt. Beachten Sie, dass diese in einem Extrapaar geschweifeter Klammern gesetzt werden müssen, um zu verhindern, dass sie BibTeX als Personennamen in ihre Bestandteile zerlegt. Verwenden Sie `shortauthor`, wenn Sie eine gekürzte Form des Namens oder eine Abkürzung für den Einsatz in Zitaten haben wollen.

```
author      = {{National Aeronautics and Space Administration}},
shortauthor = {NASA},
```

In den Standardzitierstilen wird der Kurzname in allen Zitaten ausgegeben, während der volle Name in der Bibliographie ausgegeben wird. Für kooperierende Herausgeber verwenden Sie die entsprechenden Felder `editor` und `shorteditor`. Da alle dieser Felder nun als Namenslisten behandelt werden, ist es möglich, persönliche Namen und kooperierende Namen gemischt aufzuführen, vorausgesetzt, sie sind in Klammern gesetzt.

```
editor      = {{National Aeronautics and Space Administration}
               and Doe, John},
shorteditor = {NASA and Doe, John},
```

Benutzer, die von `jurabib` zu `biblatex` gewechselt haben, sollten beachten, dass `shortauthor` als Namensliste behandelt wird.

2.3.4. Wörterlisten

Die Felder `institution`, `organization`, `publisher` und `location` sind Wörterlisten im Sinne des § 2.2. Dies gilt auch für `origlocation`, `origpublisher` und die Aliasfelder `address` and `school`. Alle diese Felder enthalten eine Liste von Elementen, die durch `,and` getrennt werden. Wenn sie ein wörtliches `,and` enthalten, muss es in Klammern gesetzt werden.

```
publisher   = {William Reid {and} Company},
institution = {Office of Information Management {and} Communications},
organization = {American Society for Photogrammetry {and}
                Remote Sensing and American Congress on Surveying
                {and} Mapping},
```

Beachten Sie den Unterschied zwischen einem wörtlichen ‚and‘ und dem Listentrennzeichen ‚and‘ in den obigen Beispielen. Sie können auch den gesamten Namen in Klammern setzen:

```
publisher    = {{William Reid and Company}},
institution  = {{Office of Information Management and Communications}},
organization = {{American Society for Photogrammetry and Remote Sensing}
                and {American Congress on Surveying and Mapping}},
```

Ältere Dateien, die noch nicht für den Einsatz mit `biblatex` aktualisiert wurden, funktionieren trotzdem, wenn diese Felder kein wörtliches ‚and‘ enthalten. Beachten Sie jedoch, dass Sie die zusätzlichen Merkmale für wörtliche Listen in diesem Fall vermissen, wie die konfigurierbare Formatierung und automatische Abkürzung.

2.3.5. Titel

Die folgenden Beispiele veranschaulichen, wie man mit den unterschiedlichen Arten von Titeln umgehen sollte. Beginnen wir mit einem fünfbändigen Werk, das als Ganzes angesehen wird:

```
@Book{works,
author    = {Shakespeare, William},
title     = {Collected Works},
volumes  = {5},
...

```

Die einzelnen Bände eines mehrbändigen Werkes haben in der Regel einen eigenen Titel. Angenommen, der vierte Band von *Collected Works* umfasst Shakespeare’s Sonette und wir beziehen uns nur auf dieses Band:

```
@Book{sonnets, author    = {Shakespeare, William},
maintitle = {Collected Works},
title     = {Sonntte},
volume    = {4},
...

```

Wenn die einzelnen Bände keinen eigenen Titel haben, gibt man den Haupttitel in `title` an und fügt eine Bandnummer hinzu:

```
@Book{sonnets,
author    = {Shakespeare, William},
title     = {Collected Works},
volume    = {4},
...

```

Im nächsten Beispiel werden wir uns auf einen Teil eines Bandes beziehen, der aber eine in sich geschlossene Arbeit mit eigenem Titel darstellt. Das jeweilige Band hat auch einen Titel, und es gibt immer noch den Haupttitel der gesamten Auflage:

```
@InBook{lear,
author      = {Shakespeare, William},
bookauthor  = {Shakespeare, William},
maintitle   = {Gesammelte Werke},
booktitle   = {Tragedies},
title       = {King Lear},
volume      = {1},
pages       = {53-159}, ...
```

Angenommen, der erste Band der *Collected Works* enthält ein nachgedrucktes Essay eines bekannten Gelehrten. Dies ist nicht die übliche Einleitung des Herausgebers, sondern eine in sich geschlossene Arbeit. *Collected Works* hat ebenso einen eigenen Herausgeber:

```
@InBook{stage,
author      = {Expert, Edward},
title       = {Shakespeare and the Elizabethan Stage},
bookauthor  = {Shakespeare, William},
editor      = {Bookmaker, Bernard},
maintitle   = {Collected Works},
booktitle   = {Tragedies},
volume      = {1},
pages       = {7-49}, ...
```

Siehe § 2.3.7 für weitere Beispiele.

2.3.6. Herausgeberfunktionen

Die Art der redaktionellen Tätigkeit eines Herausgebers ist in einem der `editor`-Felder aufgeführt (z. B. `editor`, `editora`, `editorb`, `editorc`) und sie können in dem entsprechenden `editor...type` Feld spezifiziert werden. Die folgenden Funktionen werden standardmäßig unterstützt. Die Funktion ‚`editor`‘ ist die Standardeinstellung. In diesem Fall wird `editortype` weggelassen.

- `editor` Der Hauptherausgeber. Dies ist die allgemeine redaktionelle Rolle und der Standardwert.
- `compiler` Ähnlich wie `editor`, wird aber verwendet, wenn die Aufgabe des Herausgebers hauptsächlich das Zusammenstellen ist.
- `founder` Der Gründer und Herausgeber einer Zeitschrift oder eines umfassenden Publikationsprojektes, wie eine ‚Gesammelte Werke‘-Edition oder eine lang andauernde Kolumne.

- continuator** Ein Redakteur, der die Arbeit des Gründers (*founder*) fortgesetzt, aber später durch den aktuellen Herausgeber (*editor*) ersetzt wurde.
- redactor** Ein zweiter Herausgeber, dessen Aufgabe es ist, die Arbeit zu schwärzen.
- collaborator** Eine weiterer Herausgeber oder ein Berater des Herausgebers.

Zum Beispiel, wenn die Aufgabe des Herausgebers in der Erstellung besteht, können Sie dies im entsprechenden `editortype`-Feld angeben:

```
@Collection{... ,
editor      = {Editor, Edward},
editortype  = {compiler},
... }
```

Es kann auch weitere Herausgeber neben dem Hauptherausgeber geben:

```
@Book{... ,
author      = {...},
editor      = {Editor, Edward},
editora     = {Redactor, Randolph},
editoratype = {redactor},
editorb     = {Consultant, Conrad},
editorbtype = {collaborator},
... }
```

Zeitschriften oder lang andauernde Veröffentlichungsprojekte können mehrere Generationen von Herausgebern enthalten. Zum Beispiel kann es einen Gründer zusätzlich zu dem derzeitigen Herausgeber geben:

```
@Book{... ,
author      = {...},
editor      = {Editor, Edward},
editora     = {Founder, Frederic},
editoratype = {founder},
... }
```

Beachten Sie, dass nur der `editor` in Zitaten angegeben wird und bei der Sortierung der Bibliographie. Wenn ein Eintrag typischerweise mit dem Gründer zitiert wird (und entsprechend in der Bibliographie sortiert), kommt der Gründer in das Feld `editor` und der aktuelle Editor kommt in eines der `editor...type`-Felder:

```
@Collection{... ,
editor      = {Founder, Frederic},
editortype  = {founder},
editora     = {Editor, Edward},
... }
```

Sie können weitere Funktionen durch Initialisierung und Definition eines neuen Lokalisierungsschlüssels hinzufügen, dessen Name dem Bezeichner im `editor...type`-Feld entspricht. Siehe §§ 3.7 und 4.9.1 für weitere Einzelheiten.

2.3.7. Veröffentlichungen und Zeitungsserien

Das `series`-Feld wird von herkömmlichen BibTeX-Stilen sowohl für den Haupttitel eines mehrbändigen Werkes, als auch für eine Zeitungsserie genutzt, d. h. eine lose Abfolge von Büchern vom gleichen Verlag, die sich mit dem gleichen Thema befassen oder zum gleichen Feld der Forschung gehören. Dies kann mehrdeutig sein. Dieses Paket stellt ein `maintitle`-Feld für mehrbändige Werke zur Verfügung und bietet `series` nur für Zeitungsserien. Die Ausgabe oder Nummer des Buches in der Reihe steht in diesem Fall in `number`:

```
@Book{...,
author      = {Expert, Edward},
title       = {Shakespeare and the Elizabethan Age},
series      = {Studies in English Literature and Drama},
number      = {57},
... }
```

Der `@article`-Eingabetyp nutzt auch `series`, aber auf eine andere Art und Weise. Zuerst wird ein Test durchgeführt, um festzustellen, ob der Wert des Feldes eine ganze Zahl ist. Wenn ja, wird es als eine Ordnungszahl ausgegeben. Wenn nicht, wird ein weiterer Test durchgeführt, um festzustellen, ob es ein Lokalisierungsschlüssel ist. Wenn ja, wird der lokalisierte String ausgegeben. Wenn nicht, wird der Wert ausgegeben. Betrachten Sie das folgende Beispiel für eine Zeitschrift, die in nummerierter Reihenfolge erschienen ist:

```
@Article{...,
journal     = {Journal Name},
series      = {3},
volume      = {15},
number      = {7},
year        = {1995},
... }
```

Dieser Eintrag wird als „*Journal Name*. 3rd ser. 15.7 (1995)“ ausgegeben. Einige Zeitschriften verwenden Bezeichnungen wie ‚alte Serie‘ und ‚neuen Serie‘ anstelle einer Nummer. Solche Bezeichnungen werden in `series` angegeben, entweder als Zeichenkette oder als Lokalisierungsschlüssel. Betrachten Sie das folgende Beispiel, welches die Verwendung des Lokalisierungsschlüssels `newseries` zeigt:

```
@Article{...,
journal     = {Journal Name},
series      = {newseries},
volume      = {9},
year        = {1998},
... }
```

Dieser Eintrag wird als „*Journal Name*. New ser. 9 (1998)“ ausgegeben. Siehe § 4.9.2 für eine Liste der standardmäßig definierten Lokalisierungsschlüssel.

Datumspezifikation	Formatierte Daten (Beispiele)	
	Kurzformat	Langformat
1850	1850	1850
1997/	1997–	1997–
1967-02	02/1967	February 1967
2009-01-31	31/01/2009	31st January 2009
1988/1992	1988–1992	1988–1992
2002-01/2002-02	01/2002-02/2002	January 2002–February 2002
1995-03-30/1995-04-05	30/03/1995-05/04/1995	30th March 1995–5th April 1995

Tabelle 2: Datumsspezifikationen

2.3.8. Datumsspezifikationen

Die Datumsfelder `date`, `origdate`, `eventdate` und `urldate` erfordern eine Datumsangabe im `yyyy-mm-dd` Format. Datumsbereiche werden als `yyyy-mm-dd/yyyy-mm-dd` angegeben. Teildaten werden unterstützt, wenn Komponenten nur am Ende weggelassen werden. Sie können einen unbefristeten Zeitraum angeben, indem der Bereichstrenner angegeben und das Enddatum weggelassen wird (z. B. `yyyy/`). Siehe Tabelle 2 für einige Beispiele für gültige Datenspezifikationen und die formatierten Daten, die automatisch von `biblatex` generiert werden. Das formatierte Datum ist sprachspezifisch und wird automatisch angepasst. Wenn `date` im Eingabetyp ist, nutzt `biblatex` auch die Felder `year` und `month` für die Abwärtskompatibilität des herkömmlichen BibTeX. Stilautoren sollten beachten, dass `date` oder `origdate` nur in der `bib`-Datei vorhanden sind. Alle Daten werden analysiert und in ihre Bestandteile zerlegt, sobald die `bib`-Datei verarbeitet wird. Die Datumskomponenten stehen für die Stile zur Verfügung § 4.2.4.3. Siehe diesen Abschnitt und Tabelle 7 auf Seite 156 für weitere Informationen.

2.3.9. Monats- und Journalausgaben

Das Feld `month` ist ein Integerfeld. Die Bibliographiestile wandeln, falls erforderlich, den Monat zu einer sprachabhängigen Zeichenfolge um. Aus Gründen der Abwärtskompatibilität sollten Sie die folgenden Abkürzungen mit drei Buchstaben verwenden: `jan`, `feb`, `mar`, `apr`, `may`, `jun`, `jul`, `aug`, `sep`, `oct`, `nov`, `dec`. Beachten Sie, dass diese Abkürzungen BibTeX-Zeichenketten sind und ohne Klammern oder Anführungszeichen angegeben werden. Also nicht `month={jan}` oder `month="jan"`, sondern `month=jan`. Es ist nicht möglich, einen Monat in der Form `month={8/9}` zu spezifizieren. Verwenden Sie stattdessen `date`, um Zeiträume anzugeben. Quartalszeitschriften sind in der Regel mit ‚Frühling‘ oder ‚Sommer‘ beschriftet, welche in `issue` angegeben werden sollte. Die Platzierung des `issue`-Feldes in `@article` Einträgen ähnelt und überschreibt das Feld `month`.

2.3.10. Seitennummerierung

Bei der Angabe einer Seite oder eines Seitenbereiches, entweder im Feld `pages` oder im `<postnote>`-Argument eines Zitierbefehls, ist es zweckmäßig, dass biblatex-Präfixe wie ‚p.‘ oder ‚pp.‘ automatisch hinzugefügt werden und dies tut dieses Paket in der Tat standardmäßig. Jedoch verwenden einige Werke ein anderes Seitennummerierungsschema bzw. werden nicht durch Seitenangaben, sondern durch Vers- oder Zeilennummer zitiert. Hier kommen `pagination` und `bookpagination` ins Spiel. Als Beispiel betrachten wir den folgenden Eintrag:

```
@InBook{key,
  title      = {...},
  pagination = {verse},
  booktitle  = {...},
  bookpagination = {page},
  pages      = {53--65},
  ... }
```

`bookpagination` beeinflusst die Formatierung der `pages` und `pagetotal`, die Liste der Referenzen. Seit `page` der Standard ist, wird dieses Feld im obigen Beispiel weggelassen. In diesem Fall wird der Seitenbereich als ‚pp. 53–65‘ formatiert. Angenommen, dass es bei einem Zitat aus einer Arbeit üblich ist, Verszahlen anstelle von Seitenzahlen in Zitaten zu verwenden, dann wird dies durch `pagination` widergespiegelt, welches die Formatierung des `<postnote>`-Arguments für jeden Zitierbefehl beeinflusst. Mit einem Zitat wie `\cite[17]{key}`, wird `<postnote>` als ‚v. 17‘ formatiert. Die Einstellung von `pagination` würde ‚§ 17‘ ergeben. Siehe § 3.11.3 für weitere Verwendungshinweise.

`pagination` und `bookpagination` sind Schlüsselfelder. Dieses Paket wird versuchen, ihren Wert als Lokalisierungsschlüssel zu verwenden, vorausgesetzt, dass der Schlüssel definiert wurde. Benutzen Sie immer die Singularform des Schlüssels in `bib`-Dateien, der Plural wird automatisch gebildet. Die Schlüssel `page`, `column`, `line`, `verse`, `section`, und `paragraph` sind vordefiniert, mit `page` als Standardwert. Die Zeichenfolge ‚none‘ hat eine besondere Bedeutung, wenn man `pagination` oder `bookpagination` Felder verwendet. Es unterdrückt das Präfix für den jeweiligen Eintrag. Wenn es keine vordefinierten Lokalisierungsschlüssel für die Seitennummerierung gibt, können Sie sie einfach hinzufügen. Vergleiche die Befehle `\NewBibliographyString` und `\DefineBibliographyStrings` in § 3.7. Sie müssen zwei Lokalisierungsstrings für jedes zusätzliche Seitennummerierungsschema definieren: Die Singularform (der Lokalisierungsschlüssel entspricht dem Wert des Feldes `pagination`) und die Pluralform (deren Lokalisierungsschlüssel muss den Singular plus den Buchstaben ‚s‘ enthalten). Sehen Sie die vordefinierten Schlüssel und Beispiele in § 4.9.2.

2.4. Hinweise und Vorsichtsmaßnahmen

Dieser Abschnitt enthält einige zusätzliche Hinweise zu den BibTeX-Schnittstellen des Paket. Er geht auch auf einige häufig auftretende Probleme ein.

2.4.1. Querverweise

2.4.1.1. Das `crossref`-Feld (BibTeX) Das `crossref`-Feld ist eine bequeme Möglichkeit Eltern- und Kindrelationen zwischen zwei verbundenen Einträgen herzustellen. Leider verwendet BibTeX eine Eins-zu-eins-Feld-Abbildung, welche den Nutzen des `crossref`-Feldes deutlich reduziert. Hier gibt es zwei Befehle des symmetrischen Feld-Mappings, wie das folgende Beispiel zeigt:

```
@Book{book,
author   = {Author},
bookauthor   = {Author},
title    = {Booktitle},
booktitle  = {Booktitle},
subtitle  = {Booksubtitle},
booksubtitle= {Booksubtitle},
publisher = {Publisher},
location  = {Location},
date      = {1995},
}
@InBook{inbook,
crossref  = {book},
title     = {Title},
subtitle  = {},
pages     = {5--25},
}
```

Zum Beispiel ist BibTeX nicht fähig, das `title`-Feld von den Eltern auf das `booktitle`-Feld des Kindes zu übertragen, den Titel des Buches scheint es dann zweimal zu geben. Der Stil muss `booktitle` der Eltern ignorieren, weil er nur mit den Einschränkungen von BibTeX arbeiten kann. Es ist möglich dieses Problem zu umgehen, indem man einen leeren `subtitle` in den Kindereintrag einfügt.

2.4.1.2. Das `crossref`-Feld (Biber) Mit Biber gehören die Einschränkungen bei dem `crossref`-Feld von BibTeX der Vergangenheit an. Die Einträge von Biber ermöglichen ein angemessenes Cross-Referieren mit flexiblen `data-inheritance`-Regeln. Die Duplizierung von enthaltenen Feldern im Eltern-Eintrag oder das Hinzufügen von leeren Feldern im Kindeintrag ist vergessen. Einträge werden in normaler Weise spezifiziert:

```

@Book{book,
author    = {Author},
title     = {Booktitle},
subtitle  = {Booksubtitle},
publisher = {Publisher},
location  = {Location},
date      = {1995},
}
@InBook{inbook,
crossref  = {book},
title     = {Title},
pages     = {5--25},
}

```

Das `title`-Feld der Eltern wird kopiert zum `booktitle`-Feld des Kindes, `subtitle` wird `booksubtitle`. `author` der Eltern wird `bookauthor` des Kindes, weil das Kind das `author`-Feld nicht bereitstellt; es ist auch eine Duplizierung vom Feld `author` des Kindes. Nach der Datenvererbung ist der Kindeintrag ähnlich zum Nachfolgenden:

```

author      = {Author},
bookauthor  = {Author},
title       = {Title},
booktitle   = {Booktitle},
booksubtitle = {Booksubtitle},
publisher   = {Publisher},
location    = {Location},
date        = {1995},
pages       = {5--25},

```

Siehe [appendix A](#), eine Liste von Transformationsregeln zum Herstellen des Standards. Alle sind anpassbar. Siehe [§ 4.5.3](#) für die Konfigurierung des Querverweis-Mechanismuses von Biber. Siehe auch [§ 2.2.3](#).

2.4.1.3. The `xref`-Feld Zusätzlich zum `crossref`-Feld unterstützt `biblatex` einen einfachen Querverweis-Mechanismus mit dem `xref`-Feld. Dieses ist günstig, wenn eine Relation zwischen einer Eltern-Kind-Relation bei zwei verbundenen Einträgen etabliert werden soll, die es jedoch vorzieht, ihre Unabhängigkeit zu erhalten so weit die Angaben konzentriert sind. Das `xref`-Feld unterscheidet sich vom `crossref` dahingehend, dass der Kindeintrag nicht alle Daten des Elterneintrags erben wird, die in `xref` angegeben sind. Wenn auf den Elterneintrag durch eine bestimmte Anzahl an Kindeinträgen verwiesen wird, wird `biblatex` diese automatisch in die Bibliographie einfügen. Der Schwellenwert wird durch die `mincrossrefs`-Paketoption aus [§ 3.1.2.1](#) gesteuert. Das `xref`-Feld wird von allen Backends unterstützt. Siehe dazu [§ 2.2.3](#).

2.4.2. Kapazitätsprobleme

2.4.2.1. BibTeX Eine älteres Instrument, welches ursprünglich in den 1980er Jahren entwickelt wurde, BibTeX nutzt statische Speicherzuordnungen, sehr zum Leidwesen der Anwender, die mit großen bibliographischen Datenbanken arbeiten. Mit einer großen bib-Datei, die mehrere hundert Einträge enthält, ist es sehr wahrscheinlich, dass BibTeX zu viel Speicher benötigt. Die Anzahl der Einträge, die es verkraften kann, hängt von der Anzahl der Felder, die durch BibTeX-Stil (bib) definiert wurden, ab. Stildateien, wie biblatex.bst, die eine erhebliche Anzahl der Felder definieren, können mit solchen Problemen besser umgehen. Leider gibt das herkömmliche BibTeX keine klare Fehlermeldung, wenn es zu viel Speicher benötigt, sondern gibt eine ziemlich kryptische Art von Fehlverhalten an. Die Warnmeldungen in diesem Fall sehen wie folgt aus:

```
Warning--I'm ignoring Jones1995's extra "year"
field --line 422 of file huge.bib
Warning--I'm ignoring Jones1995's extra
"volume" field --line 423 of file huge.bib
```

Diese Warnmeldungen können in der Tat darauf hinweisen, dass der Eintrag Jones1995 defekt ist, da er bereits zwei year und zwei volume Felder enthält. Ist dies nicht der Fall und ist die bib-Datei recht groß, ist es wahrscheinlich ein Kapazitätsproblem. Was diese Warnung so verwirrend macht, ist, dass sie sich nicht auf einen bestimmten Eintrag bezieht. Wenn Sie den angeblich fehlerhaften Eintrag entfernen, wird eine andere ähnliche Warnung ausgelöst. Dies ist ein Grund, warum es ratsam ist, zu bibtex8 oder Biber zu wechseln.

2.4.2.2. bibtex8 bibtex8 ist auch ein älteres Instrument und benötigt ebenso viel Arbeitsspeicher wie mit der Standardkapazität. Der Wechsel vom herkömmlichen BibTeX zu bibtex8 ist schon eine Verbesserung, weil die Kapazität des letzteren zur Laufzeit über Kommandozeile erhöht werden kann und es auch eindeutige Fehlermeldungen ausgibt, z. B.:

```
17289 fields:
Sorry---you've exceeded BibTeX's total number of fields 17250
```

Tabelle 3 gibt einen Überblick über die verschiedenen Parameter der Leistungsfähigkeit von bibtex8 und die verwendeten Befehlszeilenoptionen (Schalter), um die Standardwerte zu erhöhen. Es gibt zwei Wege, um die Kapazität der Befehlszeile zu erhöhen. Sie können eine High-Level-Änderung wie --huge verwenden, um eine Menge verschiedener Standards auszuwählen oder eine Low-Level-Änderung wählen, wie --mfields, um einen einzigen Parameter zu modifizieren. Das erste, was Sie immer tun sollten, ist bibtex8 mit --wolfgang durchlaufen zu lassen. Mit einer

Parameter	Schalter	Kapazität			
		Default	--big	--huge	--wolfgang
max_cites	--mcites	750	2000	5000	7500
max_ent_ints	--mentints	3000	4000	5000	7500
max_ent_strs	--mentstrs	3000	6000	10000	10000
max_fields	--mfields	17250	30000	85000	125000
max_strings	--mstrings	4000	10000	19000	30000
pool_size	--mpool	65530	130000	500000	750000
wiz_fn_space	--mwizfuns	3000	6000	10000	10000
hash_prime		4253	8501	16319	30011
hash_size		5000	10000	19000	35000

Tabelle 3: Kapazität und Schalter bei `bibtex8`

sehr großen Datenbank, kann jedoch auch diese Kapazität zu klein sein. In diesem Fall müssen Sie auf die Low-Level-Änderung zurückgreifen. Hier ist ein Beispiel aus einer Reihe von Änderungen, die mit einer `bib`-Datei mit ca. 1000 Einträgen bewältigt werden sollte:

```
bibtex8 --wolfgang --mcites 30000 --mentints 30000 --mentstrs 40000
--mfields 250000 --mstrings 35000 --mpool 750000 --csfile csfile.csf
auxfile
```

Bei näherer Betrachtung von Tabelle 3, werden Sie feststellen, dass es zwei Parameter gibt, die nicht direkt geändert werden können: `hash_prime` und `hash_size`. Eine Ausweitung dieser Werte ist nur mit der High-Level-Änderung möglich. Deshalb umfasst der obige Befehl daneben die Low-Level Änderung, `--wolfgang`. Diese Situation ist sehr bedauerlich, da die Hashgröße eine Obergrenze für einige andere Parameter festlegt. Zum Beispiel kann `max_strings` nicht größer als `hash_size` sein. Wenn Sie diese Grenze überschreiten, können Sie `bibtex8` mit einer größeren Kapazität kompilieren. Beachten Sie auch, dass der `wiz_fn_space`-Parameter sich nicht auf die `bib`-Datei, sondern auf den damit verbundenen Speicherbedarf der `bst`-Datei bezieht. `biblatex.bst` benötigt etwa einen Wert von etwa 6000. Der Wert 10000, der implizit durch die Verwendung `--wolfgang`, angesetzt wird, ist gut.

2.4.2.3. Biber Biber eliminiert alle der oben genannten Einschränkungen.

2.4.3. Sortier- und Kodierungsprobleme

2.4.3.1. BibTeX Herkömmliches BibTeX kann nur alphabetisierte Ascii-Zeichen korrekt. Wenn bibliographische Daten Nicht-ASCII-Zeichen enthalten, müssen diese in

Ascii-Notation angegeben werden. Zum Beispiel, anstatt den Buchstaben ‚ä‘ direkt einzugeben, muss man ihn als Eingabe der Form `\"a` mit einem Backslash und dem Ascii-Buchstaben eingeben. Die Ascii-Notation muss in geschweifte Klammern gesetzt werden. Herkömmliches BibTeX ignoriert daraufhin den Akzent und verwendet den Ascii-Buchstaben für die Sortierung. Hier ein paar Beispiele:

```
author      = {S{\'a}nchez, Jos{\}},
editor      = {Ma{\ss}mann, R{\u}diger},
translator  = {Ferd{\e}re, Fr{\c{c}}ois},
title      = {\OE}uvres compl{\e}tes,
```

Abgesehen von der unbequemen Handhabung, gibt es zwei große Probleme mit dieser Konvention. Ein kleines Problem ist, dass die zusätzlichen Klammern das Kerning von allen Nicht-Ascii-Buchstaben unterdrücken. Aber einfach alle Akzente wegzulassen, wäre nicht der richtige Weg. Zum Beispiel, im Dänischen ist der Buchstabe ‚å‘ einer der letzten Buchstaben des Alphabets, also sollte er nach ‚z‘ eingeordnet werden. BibTeX sortiert ihn jedoch wie ein ‚a‘ ein. Die Ligatur ‚æ‘ und der Buchstaben ‚ø‘ werden in dieser Sprache auch nach ‚z‘ einsortiert. Es gibt ähnliche Fälle im Norwegischen. In Schweden ist der Buchstabe ‚ö‘ einer der letzten Buchstaben des Alphabets und die Buchstaben ‚å‘ und ‚ä‘ werden ebenfalls alphabetisch nach ‚z‘, anstatt wie ein ‚a‘ eingeordnet. Besser ist hier auch die Sortierung von Ascii-Zeichen in einer eher eigentümliche Weise, da hier der Sortieralgorithmus den Ascii Zeichensatz (0-9, A-Z, a-z) verwendet wird. Dies bedeutet, dass der Kleinbuchstabe ‚a‘ nach dem Großbuchstaben ‚Z‘ kommt, was nicht nur in der Sprache, für die BibTeX ursprünglich entwickelt wurde, akzeptierbar ist. Die herkömmliche `bst`-Datei umgeht dieses Problem, indem sie alle Zeichenketten für die Sortierung in Kleinbuchstaben umwandelt, d. h. das Sortieren ist bzgl. der Groß- und Kleinschreibung wirksam. Siehe auch § 2.4.3.4.

2.4.3.2. `bibtex8` Die Umstellung auf `bibtex8` wird in solchen Fällen helfen. `bibtex8` kann nach Groß- und Kleinschreibung sortieren, und es kann 8-Bit-Zeichen korrekt behandeln, sofern es mit einer geeigneten `csf`-Datei versorgt wird und die `--csfile`-Einstellung auf der Kommandozeile angegeben wird. Dies impliziert auch, dass es möglich ist, sprachspezifische Sortierregeln auf das Literaturverzeichnis anzuwenden. Das Paket `biblatex` enthält `csf`-Dateien für einige gängige westeuropäische Kodierungen. `bibtex8` beinhaltet auch ein paar `csf`-Dateien. Beachten Sie, dass eine `biblatex.bst` nicht erkannt wird, wenn sie unter dem herkömmlichen BibTeX oder `bibtex8` läuft, daher die `bibtex8`-Paketoption. Standardmäßig ist die Sortierung nach Groß- und Kleinschreibung, da dies für das herkömmliche BibTeX erforderlich ist. Wenn die `bibtex8`-Paketoption aktiviert ist, ist die Sortierung nach Groß- und Kleinschreibung möglich.

Seit `bibtex8` abwärtskompatibel zu herkömmlichen BibTeX ist, ist es möglich, 8-Bit-Eingaben-und Ascii-Notationen zu mischen. Dies ist nützlich, wenn die Kodie-

rung in der bib-Datei nicht alle benötigten Zeichen abdeckt. Es gibt auch ein paar Grenzfälle, in denen das Ascii-Zeichen-Schema bessere Sortierergebnisse erzielt. Ein typisches Beispiel ist die Ligatur ‚œ‘. `bibtex8` behandelt diese Ligatur wie ein einzelnes Zeichen. Je nach Sortierregel, die in der `csf`-Datei definiert wurde, könnte es wie ein ‚o‘ behandelt werden oder alphabetisch nach dem Buchstaben ‚o‘ einsortiert werden, aber es kann nicht als ‚œ‘ sortiert werden. Die Ascii-Notation (`\oe`) ist äquivalent zu ‚œ‘ bei der Sortierung:

```
title      = {Œuvres complètes},
title      = {\OE}uvres complètes},
```

Manchmal ist auch das nicht ausreichend und weitere Tricks sind erforderlich. Zum Beispiel ist der Buchstabe ‚ß‘ in deutscher Sprache besonders heikel. Dieser Buchstabe wird im wesentlichen als ‚ss‘ alphabetisiert, aber nach ‚ss‘ eingeordnet. Der Name ‚Baßmann‘ würde wie folgt alphabetisch geordnet: Basmann/Bassmann/Baßmann/Bastmann. In diesem Fall wird die Ascii-Notation (`\ss`) etwas bessere Sortierergebnisse erzielen, als ‚ß‘ in Verbindung mit einer `csf`-Datei, die ‚ß‘ wie ‚s‘ behandelt:

```
author     = {Ba{\ss}mann, Paul},
```

Um es absolut richtig zu machen, muss man das Feld `sortname` nutzen:

```
author     = {Baßmann, Paul},
sortname   = {Basszzmann, Paul},
```

Nicht nur BibTeX, sondern auch LaTeX muss über die Kodierung gut Bescheid wissen. Siehe § 2.4.3.4, wie man Kodierungen angibt.

2.4.3.3. Biber Biber kann mit Ascii, 8-Bit-Kodierungen und Latin 1 und UTF-8 umgehen. Es verfügt über echte Unicodunterstützung und ist in der Lage, im laufenden Betrieb eine bib-Datei zu kodieren. Für die Sortierung verwendet Biber eine Perl-Implementierung des Unicode Collation Algorithm (UCA), wie in Unicode Technical Standard #10 dargestellt.¹² Sortierungsüberschneidungen auf der Basis von Unicode Common Locale Data Repository (CLDR) werden hinzugefügt.¹³ Die Quintessenz ist, dass Biber in vielen Fällen weit überlegene Sortierresultate liefert, sowohl für BibTeX also auch `bibtex8`. Wenn Sie an den technischen Details interessiert sind, gibt Ihnen Abschnitt 1.8 des Unicode Technical Standard #10 eine sehr kurze Zusammenfassung, warum die Unzulänglichkeiten des herkömmlichen BibTeX und sogar `bibtex8` von sehr allgemeiner Natur und nicht auf die fehlende UTF-8 Unterstützung begrenzt sind.¹⁴

¹²<http://unicode.org/reports/tr10/>

¹³<http://cldr.unicode.org/>

¹⁴http://unicode.org/reports/tr10/#Common_Misperceptions

Unicode-Unterstützung bedeutet wesentlich mehr als UTF-8-Eingabe. Unicode ist eine komplexe Standardabdeckung, die mehr als seine bekannten Teile enthält, die Unicode- Zeichenkodierung und Transportkodierungen wie UTF-8. Außerdem standardisiert es Aspekte wie Zeichensortierung, die für die Groß- und Kleinschreibung basierte Sortierung erforderlich ist. Beispielsweise kann Biber durch Verwendung des Unicode Collation Algorithm, den Buchstaben 'ß' (Beispiel in § 2.4.3.2) ohne manuellen Eingriff handhaben. Alles was Sie tun müssen, um lokalisierte Sortierung zu erhalten, ist lokal angeben:

```
\usepackage[backend=biber,sortlocale=de]{biblatex}
```

Wenn die Sprache des Dokuments mit dem Hauptgebiet des Betriebssystems zusammenhängt, wird `sortlocale` weggelassen, da Biber die lokalen Einstellungen der Umwelt automatisch erkennt. Beachten Sie, dass dies auch mit 8-Bit-Kodierungen, wie Latin 9 funktioniert, d. h., dass Sie auch einen Vorteil aus Unicode-basierter Sortierung ziehen, wenn Sie nicht mit UTF-8-Eingaben arbeiten. Siehe § 2.4.3.4, wie Eingabe- und Datenkodierungen richtig angegeben werden.

2.4.3.4. Kodierungsfestlegungen Bei Verwendung einer Nicht-Ascii-Kodierung in der `bib`-Datei, ist es wichtig zu verstehen, was `biblatex` für Sie tun kann und was einen manuellen Eingriff erfordert. Das Paket kümmert sich um die LaTeX-Seite, d. h. es wird sichergestellt, dass die Daten aus der `bb1`-Datei richtig importiert werden, sofern sie die `bibencoding`-Paketooption richtig eingestellt haben. Abhängig von der Abwärtskompatibilität kann auch die BibTeX-Seite Aufmerksamkeit fordern. Bei Verwendung von `bibtex8`, müssen Sie `bibtex8` mit einer passenden `csf`-Datei liefern, wie es über die Kodierung der `bib`-Datei bekannt ist, um in der Lage zu sein, die Einträge korrekt alphabetisch zu sortieren. Leider gibt es keine Möglichkeit für `biblatex`, diese Informationen automatisch an `bibtex8` weiterzugeben. Der einzige Weg ist `--csfile` in der Kommandozeile einzustellen, wenn Sie `bibtex8` durchlaufen lassen. Bei der Verwendung von Biber wird all dies automatisch gehandhabt und es sind keine weiteren Schritte, abgesehen von der Einstellung `bibencoding`, erforderlich. Hier sind ein paar typische Einsatzmöglichkeiten zusammen mit den relevanten Zeilen aus der Dokumentpräambel:

- Ascii-Notation sowohl in der `tex`- und `bib`-Datei mit pdfTeX oder herkömmlichen TeX (dies wird mit BibTeX, `bibtex8` und Biber funktionieren):

```
\usepackage{biblatex}
```

- Latin 1-Kodierung (ISO-8859-1) in `tex`-Dateien, Ascii-Zeichen in `bib`-Dateien mit pdfTeX oder herkömmlichen TeX (BibTeX, `bibtex8`, Biber):

```
\usepackage[latin1]{inputenc}
\usepackage[bibencoding=ascii]{biblatex}
```

- Latin 9-Kodierung (ISO-8859-15) in tex- und bib-Dateien mit pdfTeX oder herkömmlichen TeX (bibtex8, Biber):

```
\usepackage[latin9]{inputenc}
\usepackage[bibencoding=auto]{biblatex}
```

Seit `bibencoding=auto` die Standardeinstellung ist, wird die Option weglassen. Das folgende Setup hat den gleiche Effekt:

```
\usepackage[latin9]{inputenc}
\usepackage{biblatex}
```

- UTF-8-Kodierung in tex-Dateien, Latin 1 (ISO-8859-1) in bib-Dateien mit pdfTeX oder herkömmlichen TeX (bibtex8, Biber):

```
\usepackage[utf8]{inputenc}
\usepackage[bibencoding=latin1]{biblatex}
```

Das Gleiche mit XeTeX oder LuaTeX im herkömmlichen UTF-8 Modus:

```
\usepackage[bibencoding=latin1]{biblatex}
```

- Die Verwendung von UTF-8 ist sowohl in tex- als auch in bib-Dateien nicht mit dem herkömmlichen BibTeX oder bibtex8 möglich, da keines von ihnen UTF-8 behandeln kann. Sofern Sie zu Biber wechseln, müssen Sie eine 8-Bit-Codierung verwenden, wie Latin 1 (siehe oben) oder in diesem Fall auf Ascii-Notation ausweichen:

```
\usepackage[utf8]{inputenc}
\usepackage[bibencoding=ascii]{biblatex}
```

Das Gleiche mit XeTeX oder LuaTeX im herkömmlichen UTF-8-Modus:

```
\usepackage[bibencoding=ascii]{biblatex}
```

Biber kann mit der Ascii-Notation, 8-Bit-Kodierungen, wie Latin 1 und UTF-8, umgehen. Es ist auch in der Lage, im Betrieb die bib-Datei zu kodieren (ersetzt das begrenzte Kodierfeature von biblatex auf der Makroebene). Dies geschieht automatisch, falls erforderlich, vorausgesetzt, dass Sie die Kodierung der bib-Dateien korrekt angegeben haben. Zusätzlich zu den oben beschriebenen Szenarien kann Biber auch mit den folgenden Fällen umgehen:

- Klarer UTF-8-Arbeitsablauf, d. h., UTF-8 Kodierung sowohl in der `tex`- und in der `bib`-Datei mit pdfTeX oder herkömmlichen TeX:

```
\usepackage[utf8]{inputenc}
\usepackage[bibencoding=auto]{biblatex}
```

Seit `bibencoding=auto` die Standardeinstellung ist, wird diese Option weggelassen:

```
\usepackage[utf8]{inputenc}
\usepackage{biblatex}
```

Das Gleiche mit XeTeX oder LuaTeX im herkömmlichen UTF-8-Modus:

```
\usepackage{biblatex}
```

- Es ist sogar möglich, eine 8-Bit kodierte `tex`-Datei mit UTF-8 in einer `bib`-Datei zu kombinieren, sofern alle Zeichen in der `bib`-Datei auch von der 8-Bit-Codierung abgedeckt sind:

```
\usepackage[latin1]{inputenc}
\usepackage[bibencoding=utf8]{biblatex}
```

Einige Workarounds sind erforderlich bei der Verwendung des herkömmlichen TeX oder pdfTeX mit der UTF-8-Kodierung, weil der Stil `inputenc` mit `utf8`-Modulen nicht alles von Unicode abdeckt. Vergrößert gesagt, es deckt nur den westeuropäischen Unicodeteil ab. Beim Laden von `inputenc` mit `utf8`-Möglichkeiten kann `biblatex` Biber normalerweise anweisen, die `bib`-Datei zu „reencoden“. Dies kann in der `inputenc`-Datei zu Fehlern führen, wenn einige der Eigenschaften in der `bib`-Datei außerhalb des Unicodebereichs sind, der von `inputenc` supportet wird.

- Wenn Sie von diesem Problem betroffen sind, versuchen Sie es mit der `safeinputenc`-Option.

```
\usepackage[utf8]{inputenc}
\usepackage[safeinputenc]{biblatex}
```

Wenn diese Option aktiviert ist, ignoriert `inputenc` die `utf8`-Option und verwendet `Ascii`. Biber wird dann versuchen, die `bib`-Daten in die `Ascii`-Notation zu konvertieren. Beispielsweise wird es § in `\k{S}` konvertieren. Diese Option ähnelt dem Setzen von `texencoding=ascii`, sie wird aber erst in einem speziellen Szenario wirksam (`inputenc/inputenx` mit UTF-8). Dieser Workaround nutzt die Tatsache, dass sowohl Unicode als auch die UTF-8-Transport-Kodierung abwärtskompatibel zu `Ascii` sind.

Diese Lösung kann als Workaround akzeptabel sein, wenn die Daten in der bib-Datei hauptsächlich in Ascii kodiert sind. Meist sind es aber nur wenige Strings, wie einige Autorennamen, die Probleme verursachen. Man sollte jedoch im Hinterkopf behalten, dass das traditionelle TeX oder pdfTeX Unicode nicht unterstützt. Es kann helfen, wenn okkasionelle, seltsame Eigenschaften von inputenc nicht unterstützt werden, diese mit TeX zu verarbeiten, so wenn Sie einen Akzentbefehl benötigen (z. B. `\d{S}` anstelle von `Ş`). Wenn Sie vollständige Unicodeunterstützung benötigen, sollten Sie zu XeTeX oder LuaTeX wechseln.

2.4.4. Editoren- und Compilerskripts

Bei diesem Abschnitt besteht die Notwendigkeit, ihn zu aktualisieren und umzuschreiben, ihn der neuen Skriptoberfläche von biblatex anzupassen. Zur Zeit finden Sie einiges in der Dokumentation des logreq-Paketes¹⁵ und der Biblatex-Developer-Wiki.¹⁶

¹⁵<http://www.ctan.org/tex-archive/macros/latex/contrib/logreq/>

¹⁶http://sourceforge.net/apps/mediawiki/biblatex/index.php?title=Workflow_Automation

3. Benutzerhandbuch

Dieser Teil des Handbuches gibt einen Überblick über die Bedienoberfläche des `biblatex`-Pakets. Dabei deckt er alles ab, was Sie über die dem Paket beiliegenden Standardstile des `biblatex`-Pakets wissen müssen. In jedem Fall sollte diese Anleitung als erstes gelesen werden. Möchten Sie eigene Literaturverweisregeln oder Bibliographiestile erstellen, können sie mit der anschließenden Anleitung für Autoren fortfahren.

3.1. Paketoptionen

Alle Paketoptionen werden als $\langle key \rangle = \langle value \rangle$ -Notationen dargestellt. Dabei kann der Befehl `true` für alle Booleschen Schlüssel weggelassen werden. Das heißt, dass man zum Beispiel bei `sortcites` den Wert weglassen kann, was dann `sortcites=true` entspricht.

3.1.1. Paketoptionen

Die folgenden Optionen müssen gesetzt sein, während `biblatex` geladen wird. Dabei werden sie in die optionalen Argumente von `\usepackage` geschrieben.

`style=\langle file \rangle` default: numeric

Lädt den Bibliographiestil `file.bbx` sowie die Literaturverweisregel `file.cbx`. Einen Überblick zu den Standardstilen finden Sie in § 3.3.

`bibstyle=\langle file \rangle` default: numeric

Lädt den Bibliographiestil `file.bbx`. Einen Überblick zu den Standard-Bibliographiestilen finden Sie in § 3.3.2.

`citestyle=\langle file \rangle` default: numeric

Lädt die Literaturverweisregel `file.cbx`. Einen Überblick zu den Standard-Literaturverweisregeln finden Sie in § 3.3.1.

`natbib=true, false` default: false

Lädt das Kompatibilitätsmodul, welches Pseudonyme für die Befehle der Literaturverweisregeln des `natbib`-Pakets zur Verfügung stellt. Für weitere Details siehe § 3.6.9.

`mcite=true, false` default: false

Lädt ein Literaturverweisregelmodul, das Befehle zur Verfügung stellt, die ähnlich wie bei `mcite/mciteplus` funktionieren. Für Details siehe § 3.6.10.

3.1.2. Präambeloptionen

3.1.2.1. Allgemeines

Die nachfolgenden Optionen können Argumente zu `\usepackage` sein oder aber auch in der Konfigurationsdatei oder der Präambel stehen. Die Standardwerte, die unten aufgelistet sind, sind auch die Standardwerte für das Paket. Beachten Sie, dass die Bibliographie- und die Literaturverweisregelstile auch die Standardeinstellungen beim Laden verändern können. Für Details schauen Sie in § 3.3.

`sorting=nty, nyt, nyvt, anyt, anyvt, ynt, ydnt, none, debug, <name>` default: nty

Die Reihenfolge der Bibliographie: Wenn nicht anders beschrieben, dann sind die Einträge aufsteigend geordnet. Die folgenden Möglichkeiten sind standardmäßig benutzbar.

<code>nty</code>	Sortiert nach Name, Titel, Jahr.
<code>nyt</code>	Sortiert nach Name, Jahr, Titel.
<code>nyvt</code>	Sortiert nach Name, Jahr, Ausgabe, Titel.
<code>anyt</code>	Sortiert nach dem alphabetischem Etikett, Name, Jahr, Titel.
<code>anyvt</code>	Sortiert nach dem alphabetischem Etikett, Name, Jahr, Ausgabe, Titel.
<code>ynt</code>	Sortiert nach Jahr, Name, Titel.
<code>ydnt</code>	Sortiert nach Jahr (absteigend), Name, Titel.
<code>none</code>	Nicht sortieren. Alle Einträge werden in der Reihenfolge der Literaturverweise angegeben.
<code>debug</code>	Sortiert nach dem Eintragungsschlüssel. Nur für die Fehlersuche bestimmt.
<code><name></code>	Benutze <code><name></code> wie in <code>\DeclareSortingScheme</code> definiert (§ 4.5.1) Biber only

Egal welches ‚alphabetisches‘ Sortierschema man benutzt, es macht nur Sinn in Verbindung mit einem Bibliographiestil, der die entsprechenden Etiketten ausgibt. Beachten Sie, dass einige Bibliographiestile diese Paketoptionen mit einem Wert starten, der vom Paketstandard abweicht (`nty`). Für weitere Details schauen Sie in § 3.3.2. Für eine genaue Erklärung, der oben aufgelisteten Sortieroptionen so wie zum Thema Sortierprozess, schauen sie bitte in § 3.4. Um vordefinierte Schemen zu adaptieren oder neue zu definieren, schauen Sie in § 4.5.1.

`sortcase=true, false` default: true

Bestimmt, ob die Bibliographie und die Liste von Kürzeln nach Groß- und Kleinschreibung sortiert werden soll. Beachten Sie, dass die Sortierung nach Groß- und

Kleinschreibung nur von `bibtex8` und Biber-backends unterstützt wird. Sortieren nach Groß- und Kleinschreibung ist eine immer auftretende Altlast von BibTeX. Für Details beachten Sie die backend-Optionen.

`sortupper=true, false`

default: true

Biber only

Diese Option entspricht Biber's `--sortlocale` Kommandozeilen-Befehl. Sie hat keinen Effekt auf andere backend-Optionen. Ist die Option aktiviert, wird die Bibliographie nach ‚Großschreibung vor Kleinschreibung‘ sortiert. Deaktiviert, bedeutet genau anders herum.

`sortlocale=<locale>`

Biber only

Diese Option entspricht Biber's `--sortlocale` Kommandozeilen-Befehl. Sie hat keinen Effekt auf andere backend-Optionen. Ist die Option aktiviert, bestimmt sie den Ort, der zum Sortieren benutzt wird. Da `biblatex` keinen Standardwert für diese Option bereitstellt, wird Biber die lokalen Einstellungen der Umgebung übernehmen, falls kein Ort genauer spezifiziert ist.

`sortlos=bib, los`

default: los

Bestimmt die Ordnung der Liste der Kürzel. Dabei stehen folgende Möglichkeiten zur Verfügung:

- `bib` Sortieren entsprechend der Reihenfolge in der Bibliographie.
- `los` Sortiert nach Kürzel.

`sortcites=true, false`

default: false

Bestimmt, wie die Literaturverweise nach einem Literaturverweisbefehl sortiert werden, falls mehrere Schlüsseleinträge zu einem Literaturverweisbefehl zu übergeben sind. Ist diese Option aktiviert, werden die Literaturverweise nach der Reihenfolge der Bibliographie sortiert. Dieser Befehl funktioniert mit allen Literaturverweisstilen.

`maxnames=<integer>`

default: 3

Eine Grenze, die alle Namenslisten (`author`, `editor`, usw.) beeinflusst. Sollte eine Liste diese Grenze überschreiten, das heißt, wenn sie mehr als `<integer>`-Namen beinhaltet, wird sie automatisch abgeschnitten, je nach dem wie die Einstellungen der `minnames` sind.

`minnames=<integer>`

default: 1

Eine Begrenzung, die alle Namenslisten (`author`, `editor`, usw.) beeinflusst. Sollte eine Liste mehr als `<maxnames>` beinhalten, wird sie automatisch auf `<minnames>`, wie

in der Bibliographie und den Literaturverweisen, verkürzt. Das beeinflusst auch die Sortierung der Bibliographie und die Generierung der Etiketten. Der $\langle minnames \rangle$ -Wert muss natürlich kleiner oder gleich $\langle maxnames \rangle$ sein.

`maxbibnames`= $\langle integer \rangle$ default: $\langle maxnames \rangle$

Ähnlich wie `maxnames`, jedoch nur die Bibliographie beeinflussend.

`minbibnames`= $\langle integer \rangle$ default: $\langle minnames \rangle$

Ähnlich wie `minnames`, jedoch nur die Bibliographie beeinflussend.

`maxcitenames`= $\langle integer \rangle$ default: $\langle maxnames \rangle$

Ähnlich wie `maxnames`, jedoch nur die Zitierungen im Dokument beeinflussend.

`mincitenames`= $\langle integer \rangle$ default: $\langle minnames \rangle$

Ähnlich wie `minnames`, jedoch nur die Zitierungen im Dokument beeinflussend.

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

Ähnlich wie `maxnames`, aber beeinflusst alle Literaturlisten (publisher, location, usw.).

`minitems`= $\langle integer \rangle$ default: 1

Ähnlich zu `minnames`, aber beeinflusst alle Literaturlisten (publisher, location, usw.).

`autocite`=plain, inline, footnote, superscript, ...

Diese Option kontrolliert das Verhalten des `\autocite`-Befehls, wie er in § 3.6.4 behandelt wird. Die Option `plain` sorgt dafür, dass sich `\autocite` wie `\cite` verhält, `footnote` verhält sich dann wie `\footcite` und `superscript` wie `\supercite`. Die Optionen `plain`, `inline` und `footnote` sind immer verfügbar, die `superscript`-Option wird nur von den numerischen Literaturverweisstilen, welche diesem Paket beiliegen, zur Verfügung gestellt. Der Literaturverweisstil kann auch zusätzliche Optionen definieren. Die Standardeinstellungen der Option hängen vom gewählten Literaturverweisstil ab, beachten Sie dazu § 3.3.1.

`autopunct`=true, false default: true

Diese Option kontrolliert, ob die Literaturverweisbefehle nach Satzzeichen vorwärts gerichtet suchen sollen. Für Details beachten Sie § 3.6 und `\DeclareAutoPunctuation` in § 4.7.5.

Sprache	Region/Dialekt	Identifikator
Dänisch	Dänemark	danish
Dutch	Niederlande	dutch
Englisch	USA	american
	Vereinigtes Königreich	british
	Kanada	canadian
	Australien	australian
	Neu Seeland	newzealand
Finnisch	Finland	finnish
Französisch	Frankreich, Kanada	french
Deutsch	Deutsch	german
	Österreich	austrian
	Deutsch (neu)	ngerman
Deutsch (neu)	Deutschland	ngerman
	Österreich	naustrian
Griechisch	Griechenland	greek
Italienisch	Italien	italian
Norwegisch	Norwegen	norwegian
Portugiesisch	Brasilien	brazilian
	Portugal	portuguese
Spanisch	Spanien	spanish
Schwedisch	Schweden	swedish

Tabelle 4: Unterstützte Sprachen

`language=auto`, $\langle language \rangle$

default: auto

Diese Option kontrolliert die Unterstützung von Mehrsprachigkeit. Auf `auto` gestellt, wird `babel` versuchen, die Hauptsprache des Dokuments aus `babel` zu benutzen (und automatisch Englisch benutzen, sollte `babel` nicht vorhanden sein). Das ist das Standardverhalten. Es ist auch möglich, die Dokumentsprache manuell auszuwählen. Für diesen Fall hat die unten stehende `babel` Option keinen Effekt. Bitte beachten Sie Tabelle 4 für eine Liste der unterstützten Sprachen und ihren entsprechenden Identifikator.

`clearlang=true`, `false`

default: true

Wenn diese Option aktiviert ist, wird `babel` das `language`-Feld von allen Einträgen, dessen Sprache mit der `babel`-Sprache übereinstimmt, leeren (oder die Sprache, die durch die `language`-Option genauer spezifiziert ist), um sich wiederholende Spracheinträge zu vermeiden. Die Sprachkarten, die dafür benötigt werden, werden vom `\DeclareRedundantLanguages`-Befehl aus § 4.9.1 zur Verfügung gestellt.

`babel=none`, `hyphen`, `other`, `other*`

default: none

Diese Option kontrolliert die Sprachumgebung, die benutzt wird, wenn das `babel`-Paket geladen wird und ein Bibliographieeintrag ein `hyphenation`-Feld beinhaltet

(siehe: § 2.2.3). Beachten Sie, dass `biblatex` sich automatisch an die Sprache des Hauptdokuments anpasst, wenn `babel` geladen wird. In mehrsprachigen Dokumenten wird es sich auch durchgängig der aktuellen Sprache anpassen, zumindest was die Literaturverweise und die Standardsprache der Bibliographie angeht. Diese Option dient zum Wechsel der Sprache auf einem Voreintrag in der Bibliographie. Sie bietet folgende Möglichkeiten:

- `none` Schaltet die Funktion aus, d. h. es benutzt gar keine Sprachumgebung.
- `hyphen` Fügt den Eintrag in eine `hyphenrules`-Umgebung. Dies wird das Verhalten der Silbentrennung für die gewählte Sprache in das `hyphenation`-Feld des Eintrags laden, so vorhanden.
- `other` Fügt den Eintrag in eine `otherlanguage`-Umgebung. Diese wird das Verhalten der Silbentrennung für die gewählte Sprache laden, alle zusätzlichen Definitionen, welche `babel` und `biblatex` für die entsprechende Sprache zur Verfügung stellen, werden aktiviert, und Schlüssel-elemente wie `,editor‘` und `,volume‘` werden übersetzt. Diese extra Definitionen beinhalten Lokalisationen der Datenformatierung, Ordinale und ähnliche Dinge.
- `other*` Fügt den Eintrag in eine `otherlanguage*`-Umgebung. Bitte beachten Sie, dass `biblatex otherlanguage*` behandelt wie `otherlanguage`, aber andere Pakete könnten hier eine Unterscheidung machen.

`block=none, space, par, npar, ragged` default: none

Diese Option kontrolliert den extra Raum zwischen den einzelnen Einträgen, d. h. die Abstände zwischen den Bibliographieblöcken. Dabei stehen folgende Möglichkeiten zur Verfügung:

- `none` Füge nichts hinzu.
- `space` Füge zusätzlichen horizontalen Raum zwischen den Einträgen hinzu. Das hat einen ähnlichen Effekt wie das Standardverhalten der LaTeX-document-Klassen.
- `par` Beginne einen neuen Paragraphen für jeden Eintrag. Das hat einen ähnlichen Effekt wie die `openbib`-Option des Standardverhaltens der LaTeX-document-Klassen.
- `npar` Ähnlich der `par`-Option, aber verbietet Seitensprünge zwischen den Übergängen und innerhalb der Einträge.
- `ragged` Fügt eine kleine Einrückung ein, um Zeilenbrüche zwischen den Übergängen der Einträge zu fördern, außerdem wird die Bibliographie rechts „zerklüftet“ (wie bei der TextEinstellung linksbündig).

Der Befehl `\newblockpunct` kann auch verändert werden, um verschiedene Resultate zu erzielen. Beachten sie dazu auch § 4.7.1.

`notetype=foot+end, footonly, endonly` default: foot+end

Diese Option kontrolliert das Verhalten von `\mkbibfootnote`, `\mkbibendnote` und ähnliche Makros aus § 4.10.4. Die möglichen Varianten sind:

`foot+end` Unterstützt beides, Fuß- und Endnoten, d. h. `\mkbibfootnote` erstellt Fußnoten und `\mkbibendnote` Endnoten.

`footonly` Erzwingt Fußnoten, d. h. auch `\mkbibendnote` erstellt Fußnoten.

`endonly` Erzwingt Endnoten, d. h. auch `\mkbibfootnote` erstellt Endnoten.

`hyperref=true, false, auto` default: auto

Ermöglicht Literaturverweise und vice versa die Endreferenzen, zu anklickbaren Links zu machen. Benötigt das `hyperref`-Paket. Außerdem muss der gewählte Literaturverweisstil Hyperlinks unterstützen, die Standardstile aus diesem Paket tun dies bereits. Der Befehl `hyperref=auto` erkennt automatisch, ob ein `hyperref` Paket geladen wurde.

`backref=true, false` default: false

Entscheidet, ob die Endreferenzen in die Bibliographie geschrieben werden sollen. Die Endreferenzen sind eine Liste von Zahlen, die angeben, auf welchen Seiten der entsprechende Bibliographieeintrag zitiert wird. Sollten `refsection`-Umgebungen im Dokument vorhanden sein, werden die Endreferenzen an die gleiche Stelle gesetzt wie die Referenz-Einträge. Um genau zu sein, kontrolliert diese Option nur, ob das `biblatex` Paket die Daten sammeln soll, die es braucht, um solche Endreferenzen zu erzeugen. Natürlich muss auch der gewählte Bibliographiestil diese Möglichkeit unterstützen.

`backrefstyle=none, three, two, two+, three+, all+` default: three

Diese Option kontrolliert, wie die Sequenz der fortlaufenden Seiten in der Liste der Endreferenzen formatiert werden. Dabei sind folgende Stile möglich:

`none` Dieses Feature deaktivieren, d. h. die Seitenliste nicht kürzen.

`three` Jegliche Sequenz von drei oder mehr aufeinander folgenden Seiten zu einem Bereich kürzen, z. B. die Liste ‚1, 2, 11, 12, 13, 21, 22, 23, 24‘ wird zu ‚1, 2, 11–13, 21–24‘.

`two` Jegliche Sequenz von zwei oder mehr aufeinander folgenden Seiten zu einem Bereich kürzen, z. B. die obere Liste zu ‚1–2, 11–13, 21–24‘..

- two+** Ähnlich dem Konzept von `two`, aber die Sequenz von zwei genau aufeinander folgenden Seiten wird ausgegeben und benutzt die Startseite und den lokalen Folge `sequens`, z. B. wird die obere Liste zu ‚1 sq., 11–13, 21–24‘.
- three+** Ähnliche dem Konzept von `two+`, aber die Sequenz von drei genau aufeinanderfolgenden Seiten wird ausgegeben und benutzt die Startseite und den lokalen Folge `sequens`, z.B. wird die obere Liste zu ‚1 sq., 11 sq., 21–24‘.
- all+** Ähnlich dem Konzept von `three+`, aber die Sequenz aufeinander folgender Seiten wird als offener, endloser Bereich ausgegeben, z. B. wird die obere Liste zu ‚1 sq., 11 sq., 21 sq.‘ zusammengefasst.

Alle Stile unterstützten sowohl arabische als auch römische Zahlen. Um Verwechslungen bei den Listen zu vermeiden, werden verschiedene Sets von Ziffern beim Erstellen der Bereiche nicht vermischt, z. B. wird die Liste ‚iii, iv, v, 6, 7, 8‘ als ‚iii–v, 6–8‘ zusammengefasst.

`backrefsetstyle`=`setonly`, `memonly`, `setormem`, `setandmem`, `memandset`, `setplusmem` default: `setonly`

Diese Option kontrolliert, wie die Endreferenzen mit den `@set`-Einträgen und ihren Mitgliedern umgehen werden. Dabei sind die folgenden Optionen verfügbar:

- setonly** Alle Endreferenzen werden zu `@set`-Einträgen hinzugefügt. Die `pageref`-Liste der eingesetzten Mitglieder bleibt leer.
- memonly** Referenzen der eingesetzten Mitglieder werden dem entsprechenden Mitglied hinzugefügt. Referenzen der `@set`-Einträge werden zu allen Mitgliedern hinzugefügt. Die `pageref`-Liste der `@set`-Einträge bleibt leer.
- setormem** Referenzen des `@set`-Eintrags werden dem `@set`-Eintrag hinzugefügt. Referenzen zu eingesetzten Mitgliedern werden dem entsprechenden Eintrag hinzugefügt.
- setandmem** Referenzen der `@set`-Einträge werden dem `@set`-Eintrag hinzugefügt. Referenzen zu eingesetzten Mitgliedern werden dem `@set`-Eintrag und dem entsprechenden Mitglied hinzugefügt.
- memandset** Referenzen der `@set`-Einträge werden dem `@set`-Eintrag und allen Mitgliedern hinzugefügt. Referenzen der eingesetzten Mitgliedern werden dem entsprechenden Mitglied hinzugefügt.
- setplusmem** Referenzen der `@set`-Einträge werden dem `@set` Eintrag und allen Mitgliedern hinzugefügt. Referenzen der eingesetzten Mitglieder werden dem entsprechenden Mitglied und dem `@set`-Eintrag hinzugefügt.

`indexing=true, false, cite, bib` default: false

Diese Option kontrolliert die Katalogisierung in Literaturverweisen und der Bibliographie. Genauer gesagt, beeinflusst sie der `\ifciteindex`- und `\ifbibindex`-Befehl aus § 4.6.2. Die Option kann für alles, eine Pro-Typ- oder eine Pro-Eintrag-Basis gesetzt werden. Dabei stehen folgende Möglichkeiten zur Verfügung:

<code>true</code>	Aktiviert Katalogisierung für alles.
<code>false</code>	Deaktiviert Katalogisierung für alles.
<code>cite</code>	Aktiviert Katalogisierung nur für Literaturverweise.
<code>bib</code>	Aktiviert Katalogisierung nur für die Bibliographie.

Dieses Feature muss vom gewählten Literaturverweisstil unterstützt werden. Alle Standardstile, die diesem Paket beiliegen, unterstützen die Erstellung von Katalogen sowohl für Literaturverweise als auch für die Bibliographie. Beachten Sie, dass Sie immer mit `\makeindex` das Erstellen eines Katalogs aktivieren müssen, um ihn angezeigt zu bekommen.

`loadfiles=true, false` default: false

Diese Option kontrolliert, ob externe Dateien durch den `\printfile`-Befehl angefordert werden. Für Details schauen Sie auch in § 3.10.7 und `\printfile` in § 4.4.1. Beachten Sie, dass dieses Feature standardmäßig deaktiviert ist, um Leistungseinbrüche zu verhindern.

`refsection=none, part, chapter, section, subsection` default: none

Diese Option startet automatisch eine Referenzsektion an einer Dokumentenaufspaltung, wie zum Beispiel bei einem Kapitel oder einer Sektion. Das entspricht dem `\newrefsection`-Befehl. Für Details schauen Sie in § 3.5.4. Dabei stehen folgende Möglichkeiten für die Dokumentenaufspaltung zur Verfügung:

<code>none</code>	Deaktiviert dieses Feature.
<code>part</code>	Beginnt eine Referenzsektion an jedem <code>\part</code> Befehl.
<code>chapter</code>	Beginnt eine Referenzsektion an jedem <code>\chapter</code> Befehl.
<code>section</code>	Beginnt eine neue Referenzsektion an jedem <code>\section</code> Befehl.
<code>subsection</code>	Beginnt eine neue Referenzsektion an jedem <code>\subsection</code> Befehl.

Die Sternchenversionen dieser Kommandos werden keine neue Referenzsektion beginnen.

`refsegment=none, part, chapter, section, subsection` default: none

Ähnlich der `refsection`-Option, aber startet dabei ein neues Referenzsegment. Dies entspricht einem `\newrefsegment`-Befehl. Beachten Sie auch § 3.5.5 für Details.

Sollten Sie beide Optionen benutzen, beachten Sie, dass Sie diese Option nur unter einer niedrigeren Dokumentenebene als der `refsection` verwenden können, und dass verschachtelten Referenzsegmente nur lokal auf die eingekapselten Referentensektion angewandt werden.

`citereset`=none, part, chapter, section, subsection default: none

Diese Option führt automatisch den `\citereset`-Befehl von § 3.6.8 an einer Dokumentenaufspaltung, wie einem Kapitel oder einer Sektion, aus. Dabei stehen die folgenden Möglichkeiten für die Dokumentenaufteilung zur Verfügung:

- `none` Deaktivieren des Features.
- `part` Führt ein Reset an jedem `\part` Befehl aus.
- `chapter` Führt ein Reset an jedem `\chapter` Befehl aus.
- `section` Führt ein Reset an jedem `\section` Befehl aus.
- `subsection` Führt ein Reset an jedem `\subsection` Befehl aus.

Die Sternchenversionen dieser Kommandos werden kein Reset veranlassen.

`abbreviate`=true, false default: true

Entscheidet, ob lange oder verkürzte Abfolgen in den Literaturverweisen und der Bibliographie benutzt werden. Diese Option betrifft die lokalisierten Module. Sollte sie aktiviert sein, werden Schlüsseleinträge wie ‚editor‘ verkürzt. Wenn nicht, dann werden sie ausgeschrieben.

`date`=short, long, terse, comp, iso8601 default: comp

Diese Option kontrolliert das Basisformat der ausgegebenen Datenspezifikationen. Dabei stehen folgende Möglichkeiten zur Verfügung:

- `short` Benutze das kurze, wortkarge Format zur Abgrenzung:
 01/01/2010
 21/01/2010–30/01/2010
 01/21/2010–01/30/2010
- `long` Benutze das lange, wortkarge Format zur Abgrenzung:
 1st January 2010
 21st January 2010–30th January 2010
 January 21, 2010–January 30, 2010
- `terse` Benutze das kurze Format mit kompakter Abgrenzung:
 21–30/01/2010
 01/21–01/30/2010

<code>comp</code>	Benutze das lange Format mit kompakter Abgrenzung: 21st–30th January 2010 January 21–30, 2010
<code>iso8601</code>	Benutze das erweiterte ISO-8601 Format (yyyy-mm-dd): 2010-01-01 2010-01-21/2010-01-30

Wie in den Beispielen oben ersichtlich, ist das tatsächliche Format von der Sprache abhängig. Beachten Sie, dass der Monatsname in allen langen Formaten auf die `abbreviate`-Paketoption reagiert. Die vorangestellten Nullen in allen kurzen Formaten können einzeln durch die `datezeroes`-Paketoption kontrolliert werden.

`origdate=short, long, terse, comp, iso8601` default: comp

Ähnlich der `data`-Option, aber kontrolliert das Format von `origdate`.

`eventdate=short, long, terse, comp, iso8601` default: comp

Ähnlich der `date`-Option, aber kontrolliert das Format von `eventdate`.

`urldate=short, long, terse, comp, iso8601` default: short

Ähnlich der `date`-Option, aber kontrolliert das Format von `urldate`.

`alldates=short, long, terse, comp, iso8601`

Setzt alle oben genannten Datumsoptionen auf den selben Wert.

`datezeros=true, false` default: true

Diese Option kontrolliert, ob bei `short` und `terse` die Daten mit vorangestellten Nullen ausgegeben werden.

`dateabbrev=true, false` default: true

Diese Option kontrolliert ob die Daten bei `long` und `com` mit langen oder abgeschnittenen Monatsnamen ausgegeben werden. Diese Option ist ähnlich der allgemeinen `abbreviate`-Option, aber spezifiziert das Format des Datums.

`defernumbers=true, false` default: false

Im Gegensatz zum Standard-LaTeX werden durch dieses Paket die nummerierten Etiketten normalerweise an die komplette Liste der Referenzen am Anfang eines Dokuments angehängen. Sollte diese Option aktiviert sein, werden die nummerierten Etiketten (also die `labelnumber` wie in § 4.2.4) behandelt) angehängen, wenn ein Bibliographieeintrag das erste Mal, in jeglicher Bibliographie, ausgegeben wird. Für weitere Erklärungen beachten Sie auch § 3.11.5.

`punctfont=true, false`

default: false

Diese Option aktiviert einen alternativen Mechanismus, der benutzt wird, wenn eine Einheit von Satzzeichen nach einem Feld, welches in einer anderen Schriftart steht (z. B. ein Titel der kursiv geschrieben ist) genutzt wird. Für Details beachten Sie `\setpunctfont` in § 4.7.1.

`arxiv=abs, ps, pdf, format`

default: abs

Wählt einen Pfad für arXiv-Verknüpfungen. Wenn Hyperlinks-Unterstützung aktiviert ist, kontrolliert diese Option, auf welche Version des Dokuments die arXiv-Verknüpfung zeigt. Dabei stehen die folgenden Möglichkeiten zur Verfügung:

- `abs` Verlinkt den Entwurf der Seite.
- `ps` Verlinkt die PostScript Version.
- `pdf` Verlinkt die PDF Version.
- `format` Verlinkt zur Formatauswahl-Seite.

Für Details zum Support und elektronische Veröffentlichungsinformationen schauen Sie in § 3.10.6.

`backend=bibtex, bibtex8, bibtexu, biber`

default: bibtex

Spezifiziert die Datenbanken-backends. Dabei werden die folgenden „Backends“ unterstützt:

- `bibtex` Altlast von BibTeX. Traditionell unterstützt BibTeX nur die Ascii-Verschlüsselung. Dabei wird immer nach Groß- und Kleinschreibung sortiert.
- `bibtex8` `bibtex8` ist die 8-bit-Implementation von BibTeX, unterstützt Ascii- und 8-bit-Verschlüsselung. Abhängig vom `csf` kann die Sortierung nach Groß- und Kleinschreibung unterstützt werden.
- `bibtexu` `bibtexu` ermöglicht die Implementierung von Unicodes in BibTeX, welche UTF-8 unterstützen. Beachten Sie, dass `bibtexu` nicht aktiv von `biblatex` unterstützt wird und bisher nicht als Backend getestet wurde. Biber ist hier das empfohlene Backend.
- `biber` Biber ist die nächste Generation des Backend von `biblatex`-Paketen, unterstützt Ascii- und 8-bit-Verschlüsselung, UTF-8, und Re-Verschlüsselung während der Übertragung, lokalabhängiges Sortieren und vieles mehr. Sortierung nach lokaler Spezifizierung, nach Groß- und Kleinschreibung, Groß- und Klein-Priorität wird von den `sortlocale`, `sortcase`, und `sortupper` Optionen gesteuert.

Diese Option wird normalerweise immer in die Konfigurationsdatei geschrieben. Für Details beachten Sie § 3.2. Auch finden Sie in § 2.4.3 weitere Anweisungen zum Verschlüsseln von bib-Dateien.

`texencoding=auto, <encoding>` default: auto

Spezifiziert die Verschlüsselung des tex-Files. Diese Option wirkt sich auf den Datentransfer vom Backend zu biblatex aus. Bei der Verwendung von Biber entspricht diese der `--bblencoding`-Option von Biber. Zur Verfügung stehen folgende Wahlmöglichkeiten:

`auto` Versuchen Sie selbst die Eingabekodierung herauszufinden. Wenn das `inputenc/inputenx/luainputenc`-Paket verfügbar ist, erhält biblatex die wichtigste Kodierung von diesem Paket. Wenn nicht, nimmt es UTF-8-Kodierung, wenn XeTeX oder LuaTeX erkannt wurden, an und ansonsten dann Ascii.

`<encoding>` Spezifiziert das `<encoding>` explizit. Dies trifft für den seltenen Fall zu, in dem die automatische Erkennung fehlschlägt oder Sie aus einem speziellen Grund eine bestimmte Kodierung erzwingen möchten.

Beachten Sie, dass das Einstellen von `texencoding=<encoding>` sich auch auf die `bibencoding`-Option bei `bibencoding=auto` auswirkt.

`bibencoding=auto, <encoding>` default: auto

Spezifiziert die Verschlüsselung des bib-Files. Bei der Verwendung von Biber entspricht diese der `--bblencoding`-Option von Biber. Zur Verfügung stehen folgende Wahlmöglichkeiten:

`auto` Benutzen Sie diese Option, wenn der Arbeitsablauf transparent sein soll, d. h., wenn die Verschlüsselung der bib-Datei identisch zur Verschlüsselung der tex-Datei ist.

`<encoding>` Wenn das Encoding des bib-Files sich unterscheidet von dem des tex-Files, geben Sie `<encoding>` explizit an.

biblatex setzt normalerweise voraus, dass die tex- und bib-Dateien die gleiche Kodierung haben (`bibencoding=auto`). Beachten Sie, dass das gleiche Backends nur eine limitierte Anzahl von Encodings unterstützen. Sie finden in § 2.4.3 weitere Erläuterungen dazu.

`safeinputenc=true, false` default: false **Biber only**

Wenn diese Option aktiviert ist, wird automatisch `texencoding=ascii` forciert, wenn das `inputenc/inputenx`-Paket geladen wurde und die Eingabekodierung UTF-8 ist, d. h. es ignoriert alle UTF-8-basierten Makros und nimmt nur Ascii. Biber wird

dann versuchen, alle Nicht-Ascii-Daten in dem Verzeichnis bib-Datei in Ascii-Kodierung umzuwandeln. Beispielsweise wird es § in `\d{S}` konvertieren. Siehe § 2.4.3.4 für die Erläuterung, warum Sie vielleicht dies Option aktivieren.

`bibwarn=true, false` default: true

Im Normalfall wird `biblatex` automatisch Probleme mit den Daten, die Backend betreffen, wie LaTeX-Probleme übermitteln. Um dies abzustellen, benutzen Sie diese Option.

`mincrossrefs=<integer>` default: 2

Setzt das Minimum von Querverweisen auf `<integer>` wenn ein BibTeX-Lauf angefordert wird.¹⁷ Beachten Sie, dass dieses Paketooption nur die Formatierung von bestimmten Anfragen, die in die Transkriptdatei geschrieben wurden, beeinflusst. Es hat keinerlei Effekt, wenn der Editor oder Compiler, der BibTeX lädt, nicht die zugeordnete `biblatex`-Datei unterstützt, die mit BibTeX manuell über die Kommandozeile gestartet wird.¹⁸ Für Details schauen Sie in § 2.4.4. Diese Option beeinflusst auch die Handhabung des `xref`-Felds. Beachten Sie die Feld-Beschreibung in § 2.2.3 und § 2.4.1 für Details.

3.1.2.2. Stilspezifik Die folgenden Optionen werden von den Standardstilen (entgegen den Kernpaketen) zur Verfügung gestellt. Technisch gesehen sind sie Präambel-Optionen, wie die in § 3.1.2.1.

`isbn=true, false` default: true

Diese Option kontrolliert, ob die Felder `isbn/issn/isrn` ausgegeben werden.

`url=true, false` default: true

Diese Option kontrolliert, ob das `url`-Feld und die Zugangsdaten ausgegeben werden. Die Option beeinflusst nur die Einträge, dessen `url`-Informationen optional sind. Das `url`-Feld des `@online` Eintrags wird immer mit ausgegeben.

`doi=true, false` default: true

Diese Option kontrolliert, ob das `doi` mit ausgegeben wird.

`eprint=true, false` default: true

Diese Option kontrolliert, ob die `eprint`-Information mit ausgegeben wird.

¹⁷Sollte ein Eintrag, welcher von anderen Einträgen als Querverweis angegeben wird, in der `bib`-Datei auf einen Grenzwert treffen, wird er in die Bibliographie aufgenommen, selbst wenn er nicht zitiert wurde. Das ist eine Standardeigenschaft von BibTeX und nicht `biblatex` spezifisch. Beachten Sie die Beschreibung des `crossref`-Felds in § 2.2.3 für weitere Informationen.

¹⁸Wie aus diesen Worten zu entnehmen ist, ist kein LaTeX-Editor oder Compiler Skript mit zugehöriger `biblatex`-Unterstützung bekannt, aber das wird sich hoffentlich bald ändern.

3.1.2.3. Intern Die Standardeinstellungen der folgenden Prämbel-Optionen werden von der Bibliographie und den Literaturverweisstilen kontrolliert. Abgesehen von den `pagetracker` und `firstinits` Optionen, je nach dem welche Sie bevorzugen, gibt es keinen Grund diese explizit zu setzen.

`pagetracker=true, false, page, spread` default: false

Diese Option kontrolliert die Seitensuche, welche von `\ifsamepage-` und `\iffirstonpage-` Tests aus § 4.6.2 benutzt wird. Dabei stehen folgende Möglichkeiten zur Verfügung.

- `true` Aktiviert den automatischen Suchmodus. Das ist dasselbe, wie `spread`, wenn LaTeX im zweiseitigen Modus ist und bei anderen Fällen wie `page`.
- `false` Deaktiviert die Suche.
- `page` Aktiviert die Suche im Seiten-Modus. In diesem Modus funktioniert die Suche auf einer „pro-Seite“-Basis.
- `spread` Aktiviert die Suche im Spread-Modus. In diesem Modus funktioniert die Suche auf einer „Pro-Spread“-Basis.

Beachten Sie, dass diese Suche in allen Umläufen deaktiviert ist, schauen Sie in § 4.11.5.

`citecounter=true, false, context` default: false

Diese Option kontrolliert den Zitierungsschalter, der nötig ist für `citecounter` aus § 4.6.2. Dabei stehen folgende Möglichkeiten zur Verfügung.

- `true` Aktiviert den Zitierungsschalter auf globaler Ebene.
- `false` Deaktiviert den Zitierungsschalter.
- `context` Aktiviert den Zitierungsschalter im Kontext-sensitivem-Modus. In diesem Modus werden Literaturverweise in Fußnoten und im Text unabhängig von einander gesucht.

`citetracker=true, false, context, strict, constrict` default: false

Diese Option kontrolliert die Literaturverweissuche, welche erforderlich ist für den `\ifciteseen` und den `\ifentryseen` Tests aus § 4.6.2. Dabei stehen folgende Möglichkeiten zur Verfügung:

- `true` Aktiviert den Sucher auf globaler Ebene.
- `false` Deaktiviert den Sucher.
- `context` Aktiviert den Sucher im Kontext-sensitivem-Modus. In diesem Modus werden Literaturverweise in Fußnoten und im Text unabhängig von einander gesucht.

`strict` Aktiviert den Sucher im Genaukeitsmodus. In diesem Modus wird jeder Eintrag nur vom Sucher beachtet, wenn in einem allein stehendem Literaturverweis auftaucht, d.h. wenn ein einzelner Eintragungsschlüssel im Literaturverweismodus angehangen wurde.

`constrict` Dieser Modus kombiniert die Eigenschaften von `context` und `strict`.

Beachten Sie, dass dieser Sucher in allen Umläufen deaktiviert ist, schauen sie auch in § 4.11.5.

`ibidtracker=true, false, context, strict, constrict` default: false

Diese Option kontrolliert den ‚ibidem‘-Sucher, welcher vom `\ifciteibid`-Test beansprucht wird. Dabei stehen folgende Möglichkeiten zur Verfügung:

`true` Aktiviert Sucher im globalen Modus.

`false` Deaktiviert Sucher.

`context` Aktiviert den Sucher im Kontext-Sensitivem-Modus. In diesem Modus werden Literaturverweise in Fußnoten und im Text unabhängig voneinander gesucht.

`strict` Aktiviert den Sucher im genauen Modus. In diesem Modus werden potentiell mehrdeutige Verweise unterdrückt. Ein Verweis ist mehrdeutig, wenn entweder die aktuelle Zitierung (eine der ‚ibidem‘-Referierungen) aus einer Liste von Referenzen besteht.¹⁹

`constrict` Dieser Modus kombiniert Eigenschaften von `context` und `strict`. Darüber hinaus versucht der Weg der Fußnote, Zahlen und potentielle mehrdeutige Verweise in Fußnoten zu erkennen, in einer strengeren Weise als die `strict`-Option. Zusätzlich zu den Bedingungen, die die `strict`-Option auferlegt, wird ein Verweis in einer Fußnote nur als eindeutig in Betracht gezogen, wenn das aktuelle Zitat und die vorherige Nennung in der gleichen Fußnote oder in unmittelbar aufeinander folgenden Fußnoten auftreten.

Beachten Sie, dass diese Suche in allen Umläufen (floats) deaktiviert ist, schauen Sie in § 4.11.5.

`opcitracker=true, false, context, strict, constrict` default: false

Diese Option kontrolliert den ‚opcit‘-Sucher, was durch den `\ifopcit`-Test aus § 4.6.2 erforderlich ist. Diese Funktion ist ähnlich dem ‚ibidem‘-Sucher, außer dass die Zitate-

¹⁹Angenommen, das erste Zitat ist „Jones, *Title*; Williams, *Title*“ und das folgende ein „ibidem“. Aus technischer Sicht ist es ziemlich klar, dass ‚ididem‘ sich auf ‚Williams‘ bezieht, denn dies ist die letzte Referenz, die vom vorherigen Zitatbefehl verarbeitet wurde. Für einen menschlichen Leser ist dies jedoch nicht offensichtlich, weil sich ‚ibidem‘ auf beide Titel beziehen kann. Den strikten Modus sollte man bei mehrdeutigen Verweisen vermeiden.

Suche auf der author/editor-Basis erfolgt, d. h. `\ifopcit` erbringt `true`, wenn sie in der gleichen Weise erfolgt wie beim letzten Mal bei diesem author/editor. Die Möglichkeiten sind:

- `true` Aktiviert den Sucher im globalen Modus.
- `false` Deaktiviert den Sucher.
- `context` Aktiviert den Sucher im Kontext-sensitivem-Modus. In diesem Modus werden Literaturverweise in den Fußnoten und im Text separat gesucht.
- `strict` Aktiviert den Sucher im globalen Modus. In diesem Modus werden potentiell ambige Referenzen aufgehoben. Siehe zu Einzelheiten `ibidtracker=strict`.
- `constrict` Dieser Modus kombiniert die Eigenschaften von `context` und `strict`. Siehe zu Einzelheiten die Ausführungen zu `ibidtracker=constrict`.

Vermerke, dass dieser Tracker in allen Floats inaktiviert ist, siehe § 4.11.5.

`loccittracker=true, false, context, strict, constrict` default: `false`

Diese Option kontrolliert den ‚loccit‘-Sucher, welcher vom `\loccit`-Test aus § 4.6.2 benötigt wird. Diese Funktion ist ähnlich der ‚opcit‘-Suche, außerdem wird auch geprüft, ob die `⟨postnote⟩`-Argumente übereinstimmen, d. h. `\ifloccit` erbringt `true`, wenn das Zitat sich auf die gleiche Seite bezieht, wie das Zitat davor. Dabei stehen folgende Möglichkeiten zur Verfügung:

- `true` Aktiviert den Sucher im globalen Modus.
- `false` Deaktiviert den Sucher.
- `context` Aktiviert den Sucher im Kontext-sensitivem-Modus. In diesem Modus werden Literaturverweise in den Fußnoten und im Text separat gesucht.
- `strict` Aktiviert den Sucher im globalen Modus. In diesem Modus werden potentiell ambige Referenzen aufgehoben. Siehe zu Einzelheiten `ibidtracker=strict`. In Ergänzung dazu, dieser Modus überprüft auch das `⟨postnote⟩`-Argument und ist numerisch (basierend auf `\ifnumerals` aus § 4.6.2).
- `constrict` Dieser Modus kombiniert die Eigenschaften von `context` und `strict`. Siehe zu Einzelheiten die Ausführungen zu `ibidtracker=constrict`. In Ergänzung dazu, dieser Modus überprüft auch das `⟨postnote⟩`-Argument und ist numerisch (basierend auf `\ifnumerals` aus § 4.6.2).

`idemtracker=true, false, context, strict, constrict` default: false

Diese Option kontrolliert den ‚idem‘-Tracker, der erforderlich ist für den `\ifciteidem` Test aus § 4.6.2. Folgende Möglichkeiten bestehen:

- `true` Aktiviert den Sucher im globalen Modus.
- `false` Deaktiviert den Sucher.
- `context` Aktiviert den Sucher im Kontext-sensitivem-Modus. In diesem Modus werden Literaturverweise in den Fußnoten und im Text separat gesucht.
- `strict` Dies ist ein Alias for `true`, entwickelt nur, um konsistent zu sein mit den anderen Trackern. Seit dem Austausch von ‚idem‘ bekommt man Mehrdeutigkeit nicht in dergleichen Weise wie bei ‚ibidem‘ or ‚op. cit.‘, der strikte tracking-Modus wird da nicht angewendet.
- `constrict` Dieser Modus ist ähnlich mit dem von `context`, mit einer zusätzlichen Bedingung: Eine Referenz in eine Fußnote wird nur als eindeutig anerkannt, wenn das aktuelle Zitat und die vorherige Zitierung in der selben Fußnote oder in unmittelbar aufeinander folgenden Fußnoten benannt wird.

Vermerke, dass dieser Tracker in allen Floats inaktiviert ist, siehe § 4.11.5.

`parenttracker=true, false` default: true

Diese Option kontrolliert die Einschübe der Tracker, welche die Suche von ineinander verschachtelten Klammern und Einschüben vorantreibt. Dies wird von `\parentext` und `\brackettext` genommen aus § 3.6.5 sowie `\mkbibparens` und `\mkbibbrackets` aus § 4.10.4 und `\bibopenparens`, `\bibcloseparens`, `\bibopenbracket`, `\bibclosebracket` (auch in § 4.10.4).

`maxparens=<integer>` default: 3

Gibt das Maximum erlaubter geschachtelter Ebenen von Einschüben und Klammern an. Sollten Einschübe und Klammern tiefer als dieser Wert verschachtelt sein, wird `biblatex` Fehler verursachen.

`firstinits=true, false` default: false

Wenn aktiviert, dann werden alle Vor- und Mittelnamen als Initialen ausgegeben. Diese Option beeinflusst den `\iffirstinits`-Test aus § 4.6.2.

`terseinits=true, false` default: false

Diese Option kontrolliert das Format von Initialen, die von `biblatex` generiert werden. Standardmäßig fügt `biblatex` einen Punkt nach Initialen hinzu. Ist diese Option aktiviert, wird ein gedrängtes Format ohne Punkte und Leerraum benutzt. Zum

Beispiel werden die Initiale von Donald Ervin Knuth standardmäßig als ‚D. E.‘ ausgegeben und als ‚DE‘ wenn aktiviert. Mit Biber arbeitet die Option mit der Redefinierung einiger Makros, die das Format der Initiale kontrollieren. Genauer in § 3.11.4.

Biber only

`labelalpha=true, false` default: false

Ob die speziellen Felder `labelalpha` und `extraalpha` zur Verfügung gestellt werden, für Details beachten Sie § 4.2.4. Mit Biber kann diese Option auch auf Per-Typ-Basis gesetzt werden. Siehe auch `maxalphanames` und `minalphanames`.

Biber only

`maxalphanames=<integer>` default: 3

Biber only

Ähnlich zur `maxnames`-Option, aber notwendig für das Format des `labelalpha`-Feldes.

`minalphanames=<integer>` default: 1

Biber only

Ähnlich zur `minnames`-Option, aber notwendig für das Format des `labelalpha`-Feldes.

`labelnumber=true, false` default: false

Ob das spezielle Feld `labelnumber` zur Verfügung gestellt wird, für Details beachten Sie § 4.2.4. Mit Biber ist diese Option ebenfalls auf Per-Typ-Basis setzbar.

Biber only

`labelyear=true, false` default: false

Ob die speziellen Felder `labelyear` und `extrayear` zur Verfügung gestellt werden, beachten Sie § 4.2.4 für Details. Mit Biber ist diese Option auch auf Pre-Typ-Basis setzbar.

Biber only

`singletitle=true, false` default: false

Ob die Daten, die vom `\ifsingletitle` Test zur Verfügung gestellt werden, für Details schauen Sie in § 4.6.2. Mit Biber ist diese Option auch auf Pre-Typ-Basis setzbar.

Biber only

`uniquename=true, false, init, full, allinit, allfull, mininit, minfull` default: false

Biber only

Ob ein `uniquename` Zähler gesetzt werden soll, für Details schauen Sie in § 4.6.2. Dieses Feature kann individuelle Namen in der `labelname`-Liste disambiguieren. Diese Option ist auch auf Pre-Typ-Basis setzbar. Möglich sind:

`true` Ein Alias für `full`.

`false` Abschalten dieses Features.

`init` Disambiguieren von Namen mit Initialen.

<code>full</code>	Disambiguieren von Namen mit Initialen oder vollem Namen, nach Erfordernis.
<code>allinit</code>	Ähnlich wie <code>init</code> , disambiguiert jedoch alle Namen in der <code>labelname</code> -Liste mit <code>maxnames/minnames/unique</code> list.
<code>allfull</code>	Ähnlich wie <code>full</code> , disambiguiert jedoch alle Namen in der <code>labelname</code> -Liste mit <code>maxnames/minnames/unique</code> list.
<code>mininit</code>	Eine Variante von <code>init</code> , die aber disambiguiert Namen in den Listen mit identischen Familiennamen.
<code>minfull</code>	Eine Variante von <code>full</code> , die aber disambiguiert Namen in den Listen mit identischen Familiennamen.

Beachten Sie, dass sich die `unique`name-Option auf `unique`list, den `\ifsingletitle`-Test und das `extrayear`-Feld auswirkt. Siehe § 4.11.4 für weitere Details und praktische Beispiele.

`unique`list=true, false, minyear default: false **Biber only**

Ob die `unique`list-Zähler aktualisiert werden oder nicht, siehe § 4.6.2 zu Details. Dieses Feature disambiguiert die `labelname`-Liste, wenn es eine Ambiguität bekommen hat nach einem `maxnames/minnames`-Abbruch. Im Wesentlichen überschreibt es `maxnames/minnames` auf der Per-Feld-Basis. Diese Option ist auch auf Pre-Typ-Basis setzbar. Möglich sind:

<code>true</code>	Disambiguieren der <code>labelname</code> -Liste.
<code>false</code>	Abschalten dieses Features.
<code>minyear</code>	Disambiguieren der <code>labelname</code> -Liste, nur wenn die verkürzte Liste identisch ist mit einer mit dem gleichen <code>labelyear</code> . Dieser Modus eignet sich gut für <code>author-year</code> -Stile und erfordert <code>labelyear=true</code> .

Beachten Sie, dass die `unique`list-Option sich auch auswirkt auf den `\ifsingletitle`-Test und das `extrayear`-Feld. Siehe auch § 4.11.4 für weitere Details und praktische Beispiele.

3.1.3. Eintragsoptionen

Eintragsoptionen sind Paketoptionen, welche auf einer Pre-Eintrag-Basis im `options`-Feld aus § 2.2.3 angepasst werden.

3.1.3.1. Preamble/Type/Entry Options Die folgenden Optionen sind alle auf Pre-Eintrag-Basis in das `options`-Feld setzbar. Zusätzlich dazu können sie auch als optionales Argument in `\usepackage` so wie in die Konfigurationsdatei und die Dokumentenpräambel geschrieben werden. Das ist nützlich, wenn Sie das standardmäßige Verhalten global ändern wollen.

`useauthor=true, false` default: true

Ob der `author` in Etiketten benutzt und beim Sortieren beachtet wird. Dies kann nützlich sein, wenn ein Eintrag ein `author`-Feld aufweist, aber normalerweise nicht beim Autor zitiert wird, aus welchen Gründen auch immer. Wenn `useauthor=false` gesetzt wird, heißt das nicht, dass `author` ignoriert wird. Es bedeutet nur, dass `author` bei den Etiketten nicht benutzt und beim Sortieren ignoriert wird. Der Eintrag wird durch `editor` oder `editor` alphabetisch sortiert. Benutzt man die Standardstile, wird hier `author` hinter dem Titel ausgegeben. Für weitere Informationen schauen Sie in § 3.4. Bei Biber ist diese Option auch auf Per-Typ-Basis setzbar.

Biber only

`useeditor=true, false` default: true

Ob `editor` einen fehlenden `author` in den Etiketten und während des Sortierens ersetzen soll. Dies kann nützlich sein, wenn ein Eintrag eine `editor`-Feld beinhaltet, aber nicht mit dem Editor zitiert wird. Setzt man `useeditor=false` heißt das nicht, dass `editor` komplett ignoriert wird. Es heißt nur, dass `editor` `author` in den Etiketten und beim Sortieren nicht ersetzt. Der Eintrag wird dann nach `title` alphabetisiert. Benutzt man die Standardstile, so wird hier `editor` nach dem Titel ausgegeben. Für weitere Informationen schauen Sie in § 3.4. Bei Biber ist diese Option auch auf Per-Typ Basis setzbar.

Biber only

`usetranslator=true, false` default: false

Ob `translator` fehlende `author`/`editor` bei den Etiketten oder während des Sortierens ersetzen soll. Setzt man `usetranslator=true` heißt das nicht, dass `translator` `author`/`editor` überschreibt. Es bedeutet nur, dass `translator` als Rückgriff dient, sollten `author`/`editor` fehlen oder `useauthor` und `useeditor` auf `false` gesetzt sind. Mit anderen Worten, wenn Sie ein Buch lieber mit dem Übersetzer zitieren wollen, als mit dem Autor, dann müssen Sie folgende Optionen setzen: Bei Biber ist diese Option auch auf Per-Typ-Basis setzbar.

Biber only

```
@Book{..., options =
{useauthor=false,usetranslator=true},
author = {...}, translator = {...},
...}
```

Mit den Standardstilen wird `translator` standardmäßig nach dem Titel ausgegeben. Für mehr Informationen beachten Sie auch § 3.4.

`useprefix=true, false` default: false

Ob das Präfix eines Namens (`von`, `van`, `of`, `da`, `de`, `della`, usw.) berücksichtigt wird, wenn der Nachname in Literaturverweisen ausgegeben wird. Dadurch wird auch

das Sortieren und Formatieren der Bibliographie genauso wie die Erstellung einiger Etiketten beeinflusst. Ist diese Option aktiviert, stellt `biblatex` das Präfix dem Namen immer voran. So wird zum Beispiel standardmäßig Ludwig van Beethoven als „Beethoven“ zitiert und als „Beethoven, Ludwig van“ alphabetisiert. Ist diese Option jedoch aktiviert, wird er als „van Beethoven“ zitiert und als „Van Beethoven, Ludwig“ alphabetisiert. Mit Biber kann diese Option auch auf Per-Typ-Basis gesetzt werden.

Biber only

`indexing=true, false, cite, bib`

Die `indexing` Option kann auch auf Per-Typ- oder Pre-Eintrag gesetzt werden. Für Details beachten Sie § 3.1.2.1.

3.1.3.2. Typ/Eintragoptionen Die nachfolgenden Optionen sind auf Per-Eintrag-Basis in das `options`-Feld setzbar. Sie sind nicht global benutzbar.

`skipbib=true, false` default: false

Wenn diese Option aktiviert ist, wird der Eintrag aus der Bibliographie ausgeschlossen, kann aber immer noch zitiert werden. Mit Biber ist diese Option auch auf Per-Typ-Basis setzbar.

Biber only

`skiplos=true, false` default: false

Wenn diese Option aktiviert ist, wird der Eintrag aus der Abkürzungen ausgeschlossen. Er findet sich immer noch in der Bibliographie und kann auch per Abkürzung zitiert werden. Mit Biber ist diese Option auch auf Per-Typ-Basis setzbar.

Biber only

`skiplab=true, false` default: false

Ist diese Option aktiviert, dann wird `biblatex` den Einträgen keine Etiketten zuweisen. Normalerweise wird diese Option nicht benötigt. Benutzen Sie sie mit Vorsicht! Wenn sie aktiviert ist, kann `biblatex` nicht garantieren, dass Literaturverweise für die entsprechenden Einträge einzigartig sind und Literaturverweisstile, die Etiketten benötigen, könnten dabei scheitern, validierte Literaturverweise für die Einträge zu erstellen. Mit Biber ist diese Option auch auf Per-Typ-Basis setzbar.

Biber only

`dataonly=true, false` default: false

Diese Option zu setzen entspricht den `skipbib`-, `skiplos`- und `skiplab`-Optionen. Normalerweise wird diese Option nicht benötigt. Benutzen Sie sie mit Vorsicht! Mit Biber ist diese Option auch auf Per-Typ-Basis setzbar.

Biber only

3.1.4. Vererbungsoptionen

Die folgenden Vererbungsoptionen können sowohl global im optionalen Argument zu `\documentclass` oder lokal in den optionalen Argumenten zu `\usepackage` stehen:

`openbib` Diese Option wird für rückwärtige Kompatibilität mit der Standard- \LaTeX -documentclass zur Verfügung gestellt. `openbib` entspricht dabei `block=par`. Deprecated

3.2. Globale Anwenderanpassung

Unabhängig von der Möglichkeit neue Literaturverweis- und Bibliographiestile zu erstellen, gibt es zahlreiche Möglichkeiten, die dem Paket beiliegenden Stile zu verändern. Dabei werden die manuellen Veränderungen normalerweise in die Präambel geschrieben, aber es gibt auch eine Konfigurationsdatei für dauerhafte Anpassungen. Diese Datei kann auch dazu benutzt werden, die Paketoptionen auf einen Wert zu setzen, der sich vom Standardwert unterscheidet.

3.2.1. Konfigurationsdatei

Ist diese Option (Configuration File) aktiviert, wird dieses Paket die Konfigurationsdatei `biblatex.cfg` laden. Die Datei wird am Ende des Pakets gelesen, gleich nachdem die Literaturverweis- und die Bibliographiestile geladen wurden.

3.2.2. Umgebungspaketoptionen

Die load-time-Paketoption in § 3.1.1 muss im optionalen Argument zu `\usepackage` stehen. Die Paketoptionen aus § 3.1.2 können auch in der Präambel stehen. Die Optionen werden mit dem folgendem Befehl ausgelöst:

```
\ExecuteBibliographyOptions [<entry type, ...>]{<key=value, ...>}
```

Dieser Befehl kann auch in der Konfigurationsdatei benutzt werden, um die Standardeinstellungen der Paketoption zu verändern. Einige Optionen sind auch auf Per-Type-Basis setzbar. In diesem Fall spezifiziert das optionale `<entry type>`-Argument den Eintragstyp. Das `<entry type>`-Argument kann auch eine durch Kommata unterschiedene Liste von Werten sein.

3.3. Standardstile

Dieses Kapitel bietet eine kurze Beschreibung aller Bibliographie- und Literaturverweisstile, die dem `biblatex`-Paket beiliegen. Wenn Sie ihre eigenen Stile schreiben wollen, schauen Sie in § 4.

3.3.1. Zitierstile

Der Literaturverweisstil, der diesem Paket beiliegt, besitzt einige, verbreitete Zitierstile. Alle Standardstile sind auf das `shorthand`-Feld ausgerichtet und unterstützen Hyperlinks und das Erstellen von Indexen.

numeric Dieser Stil beinhaltet ein numerisches Zitierschema, ähnlich dem der bibliographischen Standardeinrichtungen von LaTeX. Er sollte in Verbindung mit den Bibliographiestilen, die für die Ausgabe der entsprechenden Etiketten in der Bibliographie zuständig sind, eingefügt werden. Er ist für einen Literaturverweis im Text bestimmt. Dieser Stil wird die folgenden Paketoptionen beim Laden setzen: `autocite=inline`, `labelnumber=true`. Dieser Stil stellt auch eine zusätzliche Präambeloption, namens `subentry`, zur Verfügung, welche die Abfertigung der Eintragungssätze beeinflussen. Ist diese Option deaktiviert, werden Literaturverweise zu einem Mitglied dieses Satzes auf den ganzen Satz verweisen. Ist er aktiviert, dann unterstützt der Stil Verweise wie „[5c]“ die nur auf einen Untereintrag in einem Satz (das dritte Beispiel) zeigen. Für Details beachten Sie die Stilbeispiele.

numeric-comp Eine kompakte Variante des `numeric`-Stils, welcher eine Liste von mehr als zwei aufeinanderfolgende Nummern in einer Reihe ausgibt. Dieser Stil ist ähnlich dem `cite`-Paket und der `sort&compress`-Option vom `natbib`-Paket im numeralen Modus. Zum Beispiel wird anstatt „[8, 3, 1, 7, 2]“ dank diesem Stil „[1–3, 7, 8]“ ausgegeben. Er ist für Literaturverweise im Text bestimmt. Dieser Stil wird die folgenden Paketoptionen beim Laden setzen:

```
autocite=inline, sortcites=true, labelnumber=true.
```

Er stellt auch eine `subentry`-Option zur Verfügung.

numeric-verb Eine wortreiche Variante der `numeric`-Stils. Der Unterschied beeinflusst die Handhabung einer Liste von Zitierungen und ist nur offensichtlich, wenn mehrere Eingabeschlüssel an einen einzelnen Zitierungsbefehl abgegeben werden. Zum Beispiel, anstelle von „[2, 5, 6]“ würde dieser Stil „[2]; [5]; [6]“ ausgeben. Er ist für im-Text-Zitierungen vorgesehen. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, labelnumber=true.
```

Er unterstützt auch die `subentry`-Option.

alphabetic Dieser Stil führt ein alphabetisches Zitierungsschema aus, ähnlich dem `alpha.bst`-Stil vom üblichen BibTeX. Die alphabetischen Labels ähneln ansatzweise dem kompakten Autor-Jahr-Stil, aber die Art, wie sie eingesetzt werden, ist dem numerischen Zitierungsschema ähnlich. Zum Beispiel, anstelle von „Jones 1995“ würde dieser Stil das Label „[Jon95]“ benutzen. „Jones and Williams 1986“ würde erstellt werden als „[JW86]“. Dieser Stil sollte im Zusammenhang mit einem alphabetischen Bibliographiestil eingesetzt werden, der die entsprechenden Labels in die Bibliographie

ausgibt. Er ist für im-Text-Zitierung vorgesehen. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, labelalpha=true.
```

alphabetic-verb Eine wortreiche Variante des `alphabetic`-Stils. Der Unterschied beeinflusst die Handhabung einer Liste von Zitierungen und ist nur offensichtlich, wenn mehrere Eingabeschlüssel an einen einzelnen Zitierungsbefehl abgegeben werden. Zum Beispiel, anstelle von „[Doe92; Doe95; Jon98]“ würde dieser Stil „[Doe92]; [Doe95]; [Jon98]“ ausgeben. Er ist für im-Text-Zitierungen vorgesehen. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, labelalpha=true.
```

authoryear Der Stil führt ein Autor-Jahr-Zitierungsschema aus. Wenn die Bibliographie zwei oder mehr Arbeiten eines Autors enthält, die alle im selben Jahr veröffentlicht wurden, wird ein Buchstabe am Jahr angehängt. Zum Beispiel würde dieser Stil Zitierungen wie „Doe 1995a; Doe 1995b; Jones 1998“ ausgeben. Dieser Stil sollte in Zusammenhang mit einem Autor-Jahr-Bibliographiestil eingesetzt werden, der die entsprechenden Label in der Bibliographie ausgibt. Er ist vorrangig für im-Text-Zitierungen vorgesehen, aber er kann auch für Zitierungen benutzt werden, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen, zur Ladezeit aus:

```
autocite=inline, labeleyear=true, uniquename=true.
```

authoryear-comp Eine kompakte Variante des `authoryear` Stils, die den Autor nur einmal ausgibt wenn nachfolgende Referenzen, die an einen einzelnen Zitierungsbefehl weitergegeben wurden, den selben Autor benutzen. Wenn sie ebenfalls dasselbe Jahr benutzen, wird das Jahr auch nur einmal ausgegeben. Zum Beispiel, anstelle von „Doe 1995b; Doe 1992; Jones 1998; Doe 1995a“ würde dieser Stil „Doe 1992, 1995a,b; Jones 1998“ ausgeben. Er ist vorrangig für im-Text-Zitierungen vorgesehen, aber er kann auch für Zitierungen benutzt werden, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, sortcites=true, labeleyear=true, uniquename=true.
```

authoryear-ibid Eine Variante des `authoryear`-Stils, die wiederholte Zitierungen mit Abkürzungen *ebenda* ersetzt, außer wenn die Zitierung die erste auf der aktuellen Seite oder Doppelseite ist oder wenn *ebenda* zweideutig im Sinne der Paketoption `ibidtracker=constrict` sein würde. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, labeleyear=true, uniquename=true, ibidtracker=constrict, pagetracker=true.
```

Dieser Stil unterstützt auch eine zusätzliche Präambeloption, mit dem Namen `ibidpage`. Für Details sehen Sie die Stilbeispiele an.

- authoryear-icom** Ein Stil der `authoryear-comp` und `authoryear-ibid` kombiniert. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:
- ```
autocite=inline, labelyear=true, uniquename=true, ibidtracker=constrict,
pagetracker=true, sortcites=true.
```
- Dieser Stil unterstützt auch eine zusätzliche Präambeloption mit dem Namen `ibidpage`. Für Details siehe die Stilbeispiele.
- authortitle** Dieser Stil führt ein einfaches Autor-Titel-Zitierungsschema aus. Er verwendet das `shorttitle`-Feld, wenn es vorhanden ist. Er ist für Zitierungen vorgesehen, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen, zur Ladezeit auf:
- ```
autocite=footnote.
```
- authortitle-comp** Eine kompakte Variante des `authortitle`-Stils, der den Autor nur einmal ausgibt, wenn nachfolgende Referenzen, die an einen einzelnen Zitierungsbefehl weitergegeben wurden, den selben Autor benutzen. Zum Beispiel, anstelle von „Doe, *Erster Titel*; Doe, *Zweiter Titel*“ würde dieser Stil „Doe, *Erster Titel*, *Zweiter Titel*“ ausgeben. Er ist für Zitierungen vorgesehen, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:
- ```
autocite=footnote, sortcites=true.
```
- authortitle-ibid** Eine Variante des `authortitle`-Stils, der wiederholte Zitierungen mit Abkürzungen ersetzt, außer wenn die Zitierungen die erste auf der aktuellen Seite oder Doppelseite ist, oder wenn *ebenda* zweideutig im Sinne der Paketoption `ibidtracker=constrict` sein würde. Er ist für Zitierungen vorgesehen, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen, zur Ladezeit auf:
- ```
autocite=footnote, ibidtracker=constrict, pagetracker=true.
```
- Dieser Stil unterstützt eine zusätzliche Präambeloption mit dem Namen `ibidpage`. Für Details siehe die Stilbeispiele.
- authortitle-icom** Ein Stil der die Eigenschaften von `authortitle-comp` und `authortitle-ibid` kombiniert. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:
- ```
autocite=footnote, ibidtracker=constrict, pagetracker=true, sortcites=true.
```
- Dieser Stil unterstützt eine zusätzliche Präambeloptionen mit dem Namen `ibidpage`. Für Details siehe die Stilbeispiele.
- authortitle-terse** Eine prägnante Variante des `authortitle`-Stils, der nur den Titel ausgibt, wenn die Bibliographie mehr als zwei Arbeiten des entsprechenden Autors/Editors beinhaltet. Dieser Stil verwendet das `shorttitle`-Feld, wenn es verfügbar ist. Er ist sowohl für

im-Text-Zitierungen als auch für Zitierungen, die in Fußnoten angegeben werden, geeignet. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, singletitle=true, uniquename=true.
```

**authortitle-tcomp** Ein Stil, der die Eigenschaften von `authortitle-comp` und `authortitle-terse` kombiniert. Dieser Stil verwendet das `shorttitle`-Feld, wenn es verfügbar ist. Er ist sowohl für im-Text-Zitierungen als auch für Zitierungen, die in Fußnoten angegeben werden, geeignet. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, sortcites=true, singletitle=true, uniquename=true.
```

**authortitle-ticom** Ein Stil der die Eigenschaften von `authortitle-icom` und `authortitle-terse` kombiniert. Anders formuliert: eine Variante des `authortitle-tcomp` Stils mit einer *ebenda* Eigenschaft. Dieser Stil ist sowohl für im-Text-Zitierungen als auch für Zitierungen, die in Fußnoten angegeben werden, geeignet. Er setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=inline, ibidtracker=constrict, pagetracker=true, sortcites=true, singletitle=true, uniquename=true.
```

Dieser Stil unterstützt eine zusätzliche Präambeloption, mit dem Namen `ibidpage`. Für Details siehe die Stilbeispiel.

**verbose** Ein wortreicher Zitierungsstil, der eine komplette Zitierung ausgibt, ähnlich mit einem Bibliographieeintrag, wenn ein Eintrag das erste Mal zitiert wird und eine kurze Zitierung danach zitiert wird. Wenn verfügbar, wird das `shorttitle`-Feld in allen kurzen Zitierungen benutzt. Wenn das `shorthand`-Feld definiert ist, wird die „Kurzschrift“ bei der ersten Zitierung eingeführt und danach als die kurze Zitierung verwendet. Dieser Stil kann ohne ein Liste von Referenzen und Abkürzungen benutzt werden, da alle bibliographischen Daten bei der ersten Zitierung bereitgestellt werden. Es ist für Zitierungen vorgesehen, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=footnote, citetracker=context.
```

Dieser Stil unterstützt eine zusätzliche Präambeloption, mit dem Namen `citepages`. Für Details siehe die Stilbeispiel.

**verbose-ibid** Eine Variante des `verbose`-Stils, der die wiederholten Zitierungen mit der Abkürzung *ebenda* austauscht, außer wenn die Zitierung die erste auf der aktuellen Seite oder Doppelseite ist, oder wenn *ebenda* zweideutig im Sinne der Paketoption `ibidtracker=strict` sein würde. Dieser Stil ist für Zitierungen vorgesehen, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

`autocite=footnote, citetracker=context, ibidtracker=constrict, pagetracker=true.`

Dieser Stil unterstützt auch zusätzliche Präambeloptionen mit den Namen `ibidpage` und `citepages`. Für Details siehe die Stilbeispiel.

**verbose-note** Dieser Stil ist ähnlich mit dem `verbose`-Stil. Bei diesem wird eine komplette Zitierung ausgegeben ähnlich einem Bibliographieeintrag, wenn ein Eintrag das erste Mal zitiert wird und eine kurze Zitierung danach zitiert wird. Im Unterschied zu dem `verbose`-Stil ist die kurze Zitierung ein Zeiger auf die Fußnote mit der kompletten Zitierung. Wenn die Bibliographie mehr als zwei Arbeiten vom entsprechenden Autor/Editor beinhaltet, schließt der Zeiger auch den Titel ein. Wenn verfügbar, wird das `shorttitle`-Feld in allen kurzen Zitierungen benutzt. Wenn das `shorthand`-Feld definiert ist, wird es wie beim `verbose`-Stil gehandhabt. Dieser Stil kann ohne eine Liste von Referenzen und Abkürzungen benutzt werden, da alle bibliographischen Daten bei der ersten Zitierung bereitgestellt werden. Er ist ausschließlich für Zitierungen vorgesehen, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

`autocite=footnote, citetracker=context, singletitle=true.`

Dieser Stil unterstützt auch zusätzliche Präambeloptionen mit den Namen `pageref` und `citepages`. Für Details siehe die Stilbeispiele.

**verbose-inote** Eine Variante des `verbose-note`, der wiederholte Zitierungen mit der Abkürzung *ebenda* austauscht, außer wenn die Zitierung die erste auf der aktuellen Seite oder Doppelseite ist, oder wenn *ebenda* zweideutig im Sinne der Paketoption `ibidtracker=strict` sein würde. Dieser Stil ist ausschließlich für Zitierungen vorgesehen, die in Fußnoten angegeben werden. Er setzt die folgenden Paketoptionen zur Ladezeit auf:

`autocite=footnote, citetracker=context, ibidtracker=constrict, singletitle=true, pagetracker=true.`

Dieser Stil unterstützt auch zusätzliche Präambeloptionen mit den Namen `ibidpage`, `pageref` und `citepages`. Für Details siehe die Stilbeispiele.

**verbose-trad1** Dieser Stil führt ein übliches Zitierungsschema aus. Er ist ähnlich dem `verbose`-Stil. Bei diesem wird eine komplette Zitierung ausgegeben ähnlich einem Bibliographieeintrag, wenn ein Eintrag das erste Mal zitiert wird und eine kurze Zitierung danach zitiert wird. Ansonsten, benutzt er die wissenschaftlichen Abkürzungen *ebenda*, *idem*, *op. cit.* und *loc. cit.* um wiederkehrenden Autoren, Titel und Seitennummern in wiederholten Zitierungen in einer spezieller Weise zu ersetzen. Wenn das `shorthand`-Feld definiert ist, wird die Abkürzung bei der ersten Zitierung eingeführt und danach als die kurze Zitierung verwendet. Dieser Stil kann ohne eine Liste von Referenzen und Kurzschriften benutzt werden, da alle bibliographischen Daten bei der ersten Zitierung bereitgestellt werden. Er ist für für Zitierungen vorgesehen,

die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=footnote, citetracker=context, ibidtracker=constrict, idemtracker=constrict, opcittracker=context, loccittracker=context.
```

Dieser Stil unterstützt auch zusätzliche Präambeloptionen mit den Namen `ibidpage`, `strict` und `citepages`. Für Details siehe die Stilbeispiele.

**verbose-trad2** Ein anderes übliches Zitierungsschema. Es ist ähnlich dem `verbose`-Stil, aber benutzt wissenschaftliche Abkürzungen wie *ebenda* und *idem* bei wiederholten Zitierungen. Im Unterschied zum `verbose-trad1`-Stil ist die Logik von der *op. cit.* Abkürzung in diesem Stil anders und *loc. cit.* wird überhaupt nicht benutzt. Es ist sogar dem `verbose-ibid` und `verbose-inote` ähnlicher als den `verbose-trad1`. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=footnote, citetracker=context, ibidtracker=constrict, idemtracker=constrict.
```

Dieser Stil unterstützt auch zusätzliche Präambeloptionen mit den Namen `ibidpage`, `strict` und `citepages`. Für Details siehe die Stilbeispiele.

**verbose-trad3** Noch ein übliches Zitierungsschema. Es ist dem `verbose-trad2` ähnlich, benutzt aber die wissenschaftliche Abkürzungen, *ebenda* and *op. cit.*, in leicht abgewandelter Weise. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
autocite=footnote, citetracker=context, ibidtracker=constrict, loccittracker=constrict.
```

Dieser Stil unterstützt auch zusätzliche Präambeloptionen mit den Namen `strict` und `citepages`. Für Details siehe die Stilbeispiele.

**reading** Ein Zitierungsstil, der zum Bibliographiestil gehört mit dem selben Namen. Er lädt lediglich den `authortitle`-Stil.

Die folgenden Zitierungsstile sind Stile für besondere Zwecke. Sie sind nicht für die Endfassung von einem Dokument vorgesehen:

**draft** Ein Konzeptstil, der die Eintragsschlüssel in Zitierungen benutzt. Der Stil setzt die folgenden Paketoptionen zur Ladezeit, auf: `autocite=plain`.

**debug** Dieser Stil gibt eher den Eintragsschlüssel aus als irgendeine Art von Label. Er ist nur für die Fehlersuche vorgesehen und setzt die folgenden Paketoptionen zur Ladezeit auf: `autocite=plain`.

### 3.3.2. Bibliographiestile

Alle Bibliographiestile, die mit diesem Paket kommen, benutzen dasselbe grundlegende Format für die verschiedenen Bibliographieeinträge. Sie unterscheiden sich nur in der Art des Labels, das in der Bibliographie ausgegeben wird und im gesamten Formatieren der Liste von Referenzen. Es gibt einen übereinstimmenden Bibliographiestil für jeden Zitierungsstil. Beachten Sie, dass manche Bibliographiestile im Folgenden nicht erwähnt werden, da sie einfach einen allgemeineren Stil laden. Zum Beispiel, der Bibliographiestil `authortitle-comp` wird den `authortitle`-Stil laden.

**numeric** Dieser Stil gibt eine numerische Bezeichnung aus, ähnlich den bibliographischen Standardfunktionen von LaTeX. Er ist für die Verwendung im Zusammenhang mit einem numerischen Zitierungsstil vorgesehen. Beachten Sie, dass das `shorthand`-Feld die vorgegebenen Labels aufhebt. Der Stil setzt die folgenden Paketooptionen zur Ladezeit auf: `labelnumber=true`.

Dieser Stil unterstützt auch eine zusätzliche Präambeloption mit dem Namen `subentry`, die das Formatieren von Eintragssets beeinflusst. Wenn diese Option aktiv ist, werden alle Mitglieder eines Sets mit einem Buchstaben markiert, der in Zitierungen benutzt werden kann, um lieber auf ein Setmitglied Bezug zu nehmen als auf das ganze Set. Für Details siehe die Stilbeispiele.

**alphabetic** Dieser Stil gibt alphabetisches Labels aus, ähnlich dem `alpha.bst`-Stil des herkömmlichen BibTeX. Er ist für die Verwendung im Zusammenhang mit einem alphabetischen Zitierungsstil vorgesehen. Beachten Sie, dass das `shorthand`-Feld die vorgegebenen Labels aufhebt. Der Stil setzt die folgenden Paketooptionen zur Ladezeit, auf: `labelalpha=true`, `sorting=anyt`.

**authoryear** Dieser Stil unterscheidet sich von den anderen Stilen darin, dass der Erscheinungstermin nicht gegen Ende des Eintrages ausgegeben wird, sondern eher nach dem Autor/Editor. Es ist für die Verwendung im Zusammenhang mit einem Autor-Jahr-Zitierungsstil vorgesehen. Wiederkehrende Autoren- und Editornamen werden durch einen Bindestrich ersetzt, außer wenn der Eintrag der erste auf der aktuellen Seite oder Doppelseite ist. Dieser Stil unterstützt eine zusätzliche Präambeloption mit dem Namen `dashed`, die diese Eigenschaft kontrolliert. Er unterstützt auch eine Präambeloption mit dem Namen `mergedate`. Für Details siehe die Stilbeispiele. Der Stil setzt die folgenden Paketooptionen zur Ladezeit, auf:

`labelyear=true`, `sorting=nyt`, `pagetracker=true`, `mergedate=true`.

**authortitle** Dieser Stil gibt überhaupt keine Labels aus. Er ist für die Verwendung im Zusammenhang mit einem Autor-Titel-Stil vorgesehen. Wiederkehrende Autoren- und Editornamen werden durch einen Bindestrich ersetzt, außer wenn der Eintrag der erste auf der aktuellen Seite oder Doppelseite ist. Dieser unterstützt auch eine Präambeloption, mit dem Namen `dashed`, die diese Eigenschaft kontrolliert. Für Details

siehe das Stilbeispiel. Der Stil setzt die folgenden Paketoptionen zur Ladezeit, auf: `pagetracker=true`.

**verbose** Dieser Stil ist dem `authortitle`-Stil ähnlich. Er unterstützt auch eine Präambeloption mit dem Namen `dashed`. Für Details siehe das Stilbeispiel. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf: `pagetracker=true`.

**reading** Dieser besondere Bibliographiestil ist für persönliche Leselisten, kommentierte Bibliographien und Anwendungen vorgesehen. Es umfasst wahlweise die Felder `annotation`, `abstract`, `library` und `file` in der Bibliographie. Wenn gewünscht, fügt er verschiedene Arten von kurzen Headers zu der Bibliographie. Dieser Stil unterstützt auch die Präambeloptionen `entryhead`, `entrykey`, `annotation`, `abstract`, `library` und `file`, die kontrollieren, ob die entsprechenden Begriffe in der Bibliographie ausgegeben werden. Für Details siehe das Stilbeispiel. Siehe auch § 3.10.7. Der Stil setzt die folgenden Paketoptionen zur Ladezeit, auf:

```
loadfiles=true, entryhead=true, entrykey=true, annotation=true, abstract=true, library=true, file=true.
```

Die folgenden Bibliographiestile sind Stile für besondere Zwecke. Sie sind nicht für die Endfassung von einem Dokument vorgesehen:

**draft** Dieser Konzeptstil umfasst die Eintragungsschlüssel in der Bibliographie. Die Bibliographie wird nach den Eintragungsschlüsseln sortiert. Der Stil setzt die folgende Paketoption zur Ladezeit auf: `sorting=debug`.

**debug** Dieser Stil gibt alle bibliographischen Daten in tabellarischer Form aus. Er ist nur für die Fehlersuche vorgesehen und setzt die folgende Paketoption zur Ladezeit auf: `sorting=debug`.

### 3.4. Sortierungsoptionen

Dieses Paket unterstützt verschiedene Sortierungsschemata für Bibliographien. Das Sortierungsschema wird ausgewählt mit der Paketoption `sorting` aus § 3.1.2.1. Abgesehen von den normalen Datenfeldern gibt es auch besondere Felder, die benutzt werden können, um das Sortieren der Bibliographie zu verbessern. Appendices ?? and B.2 geben einen Entwurf des alphabetischen Schema an, das von `biblatex` unterstützt wird. Chronologische Sortierungsschemata sind in appendix B.3 aufgelistet. Einige Erklärungen bezüglich dieser Schemata sind sortiert.

Der erste Begriff, der im Sortierungsprozess berücksichtigt wird, ist immer das `presort`-Feld des Eintrages. Wenn dieses Feld nicht definiert ist, wird `biblatex` den vorgegebenen Wert `,mm'` als eine vorsortierte Reihung benutzen. Der nächste Begriff, der berücksichtigt wird, ist das `sortkey`-Feld. Wenn dieses Feld nicht definiert ist, dient es als Hauptsortierungsschlüssel. Abgesehen vom `presort`-Feld, werden

keine weiteren Daten in diesem Fall berücksichtigt. Wenn das `sortkey`-Feld nicht definiert ist, wird das Sortieren mit den Namen fortgesetzt. Das Paket wird versuchen, die `sortname`-, `author`-, `editor`- und `translator`-Felder in dieser Reihenfolge zu benutzen. Welche Felder berücksichtigt werden, hängt von der Konfiguration der `useauthor`-, `useeditor`- und `usetranslator`-Optionen ab. Wenn alle drei aktiv sind, wird das `sortname`-Feld ebenfalls ignoriert. Beachten Sie, dass alle Namenfelder auf die Konfiguration der allgemeinen `maxnames`- und `minnames`-Optionen reagieren. Wenn kein Feld verfügbar ist, entweder weil alle undefiniert sind oder weil `useauthor`, `useeditor` und `usetranslator` deaktiviert sind, wird `biblatex` auf die `sorttitle`- und `title`-Felder als letzten Ausweg zurückgreifen. Die restlichen Begriffe sind in verschiedener Reihenfolge: Das `sortyear`-Feld, wenn definiert, oder ansonsten die ersten vier Zahlen von dem `year`-Feld; das `sorttitle`-Feld, wenn definiert oder ansonsten das `title`-Feld; das `volume`-Feld, welches zu vier Zahlen mit führenden Nullen ausgefüllt wird oder ansonsten die Reihung 0000. Beachten Sie, dass die Sortierungsschemata, die in [appendix B.2](#) dargestellt sind, einen zusätzlichen Begriff enthalten: `labelalpha` ist das Label, benutzt vom ‚alphabetischen‘ Bibliographiestil. Genau genommen ist die Reihung, die für das Sortieren benutzt wird, `labelalpha + extraalpha`. Die Sortierungsschemata in [appendix B.2](#) sind vorgesehen, nur in Zusammenhang mit alphabetischen Stilen benutzt zu werden.

Die chronologischen Sortierungsschemata, aufgeführt in [appendix B.3](#), benutzen auch die `presort`- und `sortkey`-Felder, wenn definiert. Der nächste Begriff, der betrachtet wird, ist das `sortyear`- oder das `year`-Feld, abhängig von der Verfügbarkeit. Das `ynt`-Schema zieht die ersten vier arabischen Zahlen aus den Feld. Wenn beide Felder nicht definiert sind, wird die Reihung 9999 als Rückgriffswert benutzt. Das bedeutet, dass alle Einträge ohne Jahr zum Ende der Liste gerückt werden. Das `ydnt` ist im Konzept ähnlich, sortiert aber Jahr in absteigender Reihenfolge. Wie auch beim `ynt`-Schema wird die Reihung 9999 als Rückgriffswert benutzt. Die restlichen Begriffe sind ähnlich den alphabetischen Sortierungsschemata, die oben behandelt wurden. Beachten Sie, dass das `ydnt`-Sortierungsschema die Daten nur in absteigender Reihenfolge sortieren wird. Alle anderen Begriffe werden, wie gewohnt, in aufsteigender Reihenfolge sortiert.

Das Benutzen von besonderen Feldern wie `sortkey`, `sortname` oder `sorttitle` wird normalerweise nicht benötigt. Das `biblatex`-Paket ist durchaus fähig, die erwünschte Sortierreihenfolge durch das Benutzen der Daten, die in regulären Feldern eines Eintrags gefunden wurden, zu erstellen. Man wird sie nur gebrauchen, wenn man manuell die Sortierreihenfolge der Bibliographie ändern will oder wenn irgendwelche Daten, die für das Sortieren benötigt werden, fehlen. Bitte verweisen Sie auf die Feldbeschreibung in [§ 2.2.3](#) für Details bei möglicher Verwendungen der besonderen Felder. Beachten Sie außerdem, dass das Benutzen von `bibtex8` oder `Biber` anstelle des Alt-`bibtex` dringend empfohlen wird.

## 3.5. Bibliographiebefehle

### 3.5.1. Ressourcen

`\addbibresource`[*options*]{*resource*}

Fügt eine *resource*, wie beispielsweise eine `.bib`-Datei, zu der vorgegebenen Ressourcenliste. Dieser Befehl kann nur in der Präambel benutzt werden. Er ersetzt den alten `\bibliography`-Befehl. Beachten Sie, dass Dateien mit ihren vollständigen Namen festgelegt werden müssen, einschließlich der Erweiterungen. Lassen Sie nicht die `.bib`-Erweiterung vom Dateinamen weg. Beachten Sie außerdem, dass die *resource* eine einzelne Ressource ist. Rufen Sie `\addbibresource` mehrmals auf, um mehr Ressourcen hinzuzufügen, zum Beispiel:

```
\addbibresource{bibfile1.bib}
\addbibresource{bibfile2.bib}
\addbibresource[location=remote]{http://www.citeulike.org/bibtex/group/9517}
\addbibresource[location=remote,label=lan]{ftp://192.168.1.57/~user/file.bib}
```

Da die *resource*-Reihung in einer Wort für Wort ähnlichen Weise gelesen wird, beinhaltet es möglicherweise beliebige Zeichen. Die einzige Einschränkung ist, dass jede geschweifte Klammer ausgeglichen werden muss. Die folgenden *options* sind verfügbar:

`label`=*identifier*

Weist ein Label einer Ressource zu. Der *identifier* kann an Stelle des vollständigen Ressourcenamen im optionalen Argument von `refsection` verwendet werden (siehe 3.5.4).

`location`=*location* default: local

Der Ort der Ressource. Die *location* kann entweder `local` sein für lokale Ressourcen oder `entfernt` für URLs. Entfernte Ressourcen benötigen Biber. Die Protokolle HTTP und FTP werden unterstützt. Die entfernte URL muss ein vollständiger berechtigter Pfad zu einer `bib`-Datei sein oder eine URL, die eine `bib` Datei zurückgibt.

`type`=*type* default: file

Die Typ einer Ressource. Derzeit der einzige unterstützte Typ ist `file`.

`datatype`=*datatype* default: bibtex

Der Datentyp (Format) der Ressource. Die folgenden Formate werden derzeit unterstützt:

`bibtex` BibTeX-Format.

`ris` Research Information Systems (RIS) Format.<sup>20</sup> Beachte, dass ein Biber only

|                            |                                                                                                                                  |            |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------|------------|
|                            | ID-Kennzeichen in allen RIS benötigt wird. Der ID-Wert entspricht dem Eintragungsschlüssel. RIS Unterstützung ist experimentell. |            |
| <code>\zoterordfxml</code> | Zotero RDF/XML format. Der Support für dieses Format ist experimentell. Im Biber-manual finden Sie Details dazu.                 | Biber only |
| <code>\endnotexml</code>   | EndNote XML-format. Der Support für dieses Format ist experimentell. Im Biber-manual finden Sie Details dazu.                    | Biber only |

`\addglobalbib`[*options*]{*resource*}

Dieser Befehl unterscheidet sich von `\addbibresource`, indem die *resource* zu der allgemeinen Ressourceliste hinzugefügt ist. Der Unterschied zwischen den vorgegebenen Ressourcen und den allgemeinen Ressourcen ist nur relevant, wenn es Referenzenabschnitte im Dokument gibt und die optionalen Argumente von `refsection` (§ 3.5.4) benutzt werden, um alternative Ressourcen festzulegen, die die vorgegebene Ressourceliste ersetzt. Alle allgemeinen Ressourcen werden zu allen Referenzenabschnitten hinzugefügt.

`\addsectionbib`[*options*]{*resource*}

Dieser Befehl unterscheidet sich von `\addbibresource`, indem die Ressourcen *options* registriert werden, aber die *Ressourcen* zur keiner Ressourceliste hinzugefügt werden. Dies wird nur für Ressourcen benötigt, die 1) ausschließlich in den optionalen Argumente von `refsection` (3.5.4) stehen und 2) Optionen benötigen, die abweichen von den vorgegebenen Einstellungen. In diesen Fall kommt `\addsectionbib` zum Einsatz, um die *resource* vorher zu markieren um es zu benutzen, beim Festlegen der geeigneten *Optionen* in der Präambel. Die `label`-Option kann nützlich sein um kurze Namen den Ressourcen zuzuordnen.

`\bibliography`{*bibfile*, ... }

Deprecated

Der alte Befehl für das Hinzufügen von bibliographischen Ressourcen unterstützt die umgekehrte Kompatibilität. Wie `\addbibresource` kann dieser Befehl nur in der Präambel benutzt werden und fügt Ressourcen zu der vorgegebenen Ressourceliste hinzu. Sein Argument ist eine Komma-getrennte Liste von bib-Dateien. Die `.bib`-Erweiterung kann vom Dateinamen weggelassen werden. Den Befehl mehrmals aufzurufen, um mehrere Dateien hinzuzufügen, ist erlaubt. Dieser Befehl wird missbilligt. Bitte überlegen sie stattdessen, `\addbibresource` zu benutzen.

<sup>20</sup>[http://en.wikipedia.org/wiki/RIS\\_\(file\\_format\)](http://en.wikipedia.org/wiki/RIS_(file_format))

### 3.5.2. Die Bibliographie

`\printbibliography` [*<key=value, ...>*]

Dieser Befehl gibt die Bibliographie aus. Es erfordert ein optionales Argument, welches eine Liste von Optionen ist, die in *<key>=<value>* Notation angegeben ist. Die folgenden Optionen sind verfügbar:

`env=<name>` default: bibliography/shorthands

Das ‚high-level‘-Layout der Bibliographie und die Liste der Kürzel wird von der Nachbarschaft mit `\defbibenvironment` kontrolliert. Diese Option sucht eine Nachbarschaft aus. Der *<name>* entspricht dem Identifier, der benutzt wird, wenn die Nachbarschaft mit `\defbibenvironment` definiert wird. Automatisch benutzt der `\printbibliography`-Befehl den Identifier `bibliography`; `\printshorthands` benutzt `shorthands`. Siehe auch §§ 3.5.3 and 3.5.7.

`heading=<name>` default: bibliography/shorthands

Die Bibliographie und die Liste von Kürzeln haben typischerweise eine Kapitel- oder eine Teilüberschrift. Diese Option wählt den Überschrift *<name>* aus, wie mit `\defbibheading` definiert wird. Automatisch benutzt der `\printbibliography`-Befehl die Überschrift `bibliography`; `\printshorthands` benutzt `shorthands`. Siehe auch §§ 3.5.3 and 3.5.7.

`title=<text>`

Diese Option überschreibt den vorgegebenen Titel, der von der Überschrift bereitgestellt wird, die mit der `heading`-Option ausgewählt wurde, wenn sie von der Überschriftdefinition unterstützt wird. Für Details siehe § 3.5.7.

`prenote=<name>`

Die Pränote ist ein beliebiges Stück Text, das nach der Überschrift ausgegeben wird, aber vor der Liste von Referenzen stammt. Diese Option wählt den Pränote *<name>* aus, wie in `\defbibnote` definiert. Automatisch wird keine Pränote ausgegeben. Die Notiz wird in der Standardtextschrift ausgegeben. Sie wird nicht von `\bibsetup` und `\bibfont` beeinflusst, aber sie kann ihre eigene Schriftdeklaration beinhalten. Für Details siehe § 3.5.8.

`postnote=<name>`

Die Postnote ist ein beliebiges Stück Text, das nach der Liste von Referenzen ausgegeben wird. Diese Option wählt den Postnote-*<name>* aus, wie er in `\defbibnote` definiert ist. Automatisch wird keine Postnote ausgegeben. Die Notiz wird in der

Standardtextschrift ausgegeben. Sie wird nicht von `\bibsetup` und `\bibfont` beeinflusst, aber sie kann ihre eigene Schriftdeklaration beinhalten. Für Details siehe § 3.5.8.

`section`= $\langle integer \rangle$  default: 0

Gibt nur Einträge aus, die in der Referenzsektion  $\langle integer \rangle$  zitiert wurden. Die Referenzsektionen sind nummeriert, beginnend bei 1. Alle Zitierungen, die außerhalb einer `refsection`-Nachbarschaft angegeben werden, sind der Section 0 zugeordnet. Für Details siehe § 3.5.4 und für Anwendungsbeispiele siehe § 3.10.3. Wenn eine Kette von mehreren Filtern gegeben ist, muss der `Section` Filter immer zuerst angegeben werden.

`segment`= $\langle integer \rangle$  default: 0

Gibt nur Einträge aus, die im Referenzsegment  $\langle integer \rangle$  zitiert wurden. Die Referenzsegmente sind nummeriert, beginnend bei 1. Alle Zitierungen, die außerhalb einer `refsegment`-Nachbarschaft angegeben werden, sind dem Segment 0 zugeordnet. Für Details siehe § 3.5.5 und für Anwendungsbeispiele siehe § 3.10.3.

`type`= $\langle entrytype \rangle$

Gibt nur Einträge aus, deren Eintragstyp  $\langle entrytype \rangle$  ist.

`nottype`= $\langle entrytype \rangle$

Gibt nur Einträge aus, deren Eintragstyp nicht  $\langle entrytype \rangle$  ist. Diese Option kann mehrere Male benutzt werden.

`subtype`= $\langle subtype \rangle$

Gibt nur Einträge aus, deren `entrysubtype` definiert ist und  $\langle subtype \rangle$  ist.

`notsubtype`= $\langle subtype \rangle$

Gibt nur Einträge aus, deren `entrysubtype` undefiniert ist und nicht  $\langle subtype \rangle$  ist. Diese Option kann mehrere Male benutzt werden.

`keyword`= $\langle keyword \rangle$

Gibt nur Einträge aus, deren `keywords` Feld ein  $\langle keyword \rangle$  mit einschließt. Diese Option kann mehrere Male benutzt werden.

`notkeyword`= $\langle keyword \rangle$

Gibt nur Einträge aus, deren `keywords` kein  $\langle keyword \rangle$  mit einschließt. Diese Option kann mehrere Male benutzt werden.

`category=<category>`

Gibt nur Einträge aus, die zur Kategorie `<category>` zugeteilt sind. Diese Option kann mehrere Male benutzt werden.

`notcategory=<category>`

Gibt nur Einträge aus, die nicht zur Kategorie `<category>` zugeteilt sind. Diese Option kann mehrere Male benutzt werden.

`filter=<name>`

Filtiert die Einträge mit dem Filter `<name>`, wie er mit `\defbibfilter` definiert ist. Für Details siehe § 3.5.9.

`check=<name>`

Filtiert die Einträge mit dem Check `<name>`, wie er mit `\defbibcheck` definiert ist. Für Details siehe § 3.5.9.

`prefixnumbers=<string>`

Diese Option ist nur bei numerischen Zitierungen/Bibliographiestilen anwendbar und verlangt, dass die `defernumbers`-Option von § 3.1.2.1 allgemein aktiv ist. Diese Option einzustellen wird implizit `resetnumbers` in der aktuellen Bibliographie aktiviert. Die Option weist den `<string>`, als ein Präfix, allen Einträgen in der entsprechenden Bibliographie zu. Zum Beispiel, wenn der `<string>` A ist, wird der numerische Label, der ausgegeben wird, [A1], [A2], [A3], etc. sein. Dies ist sinnvoll, für unterteilte numerische Bibliographien, bei denen jede Subbibliographie ein anderes Präfix benutzt. Der `<string>` ist für Stile im `prefixnumber`-Feld unter allen betroffenen Einträge verfügbar. Für Details siehe § 4.2.4.2.

`resetnumbers=true, false`

Diese Option ist nur bei numerischen Zitierungen/Bibliographiestilen anwendbar und verlangt, dass die `defernumbers`-Option von § 3.1.2.1 allgemein aktiv ist. Wenn aktiv, wird sie die numerischen Label zurücksetzen, die den Einträgen in der entsprechenden Bibliographie zugeteilt werden, z. B. die Nummerierung wird bei 1 neu anfangen. Benutzen Sie diese Option mit Vorsicht, da `biblatex` keine einzigartigen Labels allgemein garantieren kann, wenn sie manuell zurückgesetzt werden.

`omitnumbers=true, false`

Diese Option ist nur bei numerischen Zitierungen/Bibliographiestilen anwendbar und verlangt, dass die `defernumbers` Option von § 3.1.2.1 allgemein aktiv ist. Wenn aktiv, wird `biblatex` ein numerisches Label nicht zu dem Eintrag in der entsprechenden Bibliographie zuweisen. Dies ist nützlich, wenn eine numerischen Bibliographie

mit einer oder mehr Subbibliographien, die ein verschiedenenes Schema (z. B. Autor-Titel oder Autor-Jahr) benutzen, vermischt wird.

`\bibbysection` [*key=value, ...*]

Dieser Befehl führt automatisch Schleifen über alle Referenzensektionen aus. Dies ist äquivalent zu dem Übergeben eines `\printbibliography`-Befehls zu jeder Sektion, hat aber den zusätzlichen Vorteil vom automatischen Überspringen von Sektionen ohne Referenzen. Beachten Sie, dass `\bibbysection` für Referenzen in der Sektion 1 zu suchen beginnt. Er wird Referenzen, die außerhalb der `refsection`-Nachbarschaft angegeben werden, ignorieren, da sie zur Section 0 zugeordnet sind. Für Anwendungsbeispiele siehe § 3.10.3. Die Optionen sind eine Teilmenge von denen, die `\printbibliography` unterstützt. Gültige Optionen sind `env`, `heading`, `prenote`, `postnote`, `maxnames`, `minnames`, `maxitems`, `minitems`.

`\bibbysegment` [*key=value, ...*]

Dieser Befehl führt automatisch Schleifen über alle Referenzensegmente aus. Dies ist äquivalent, zu dem Übergeben eines `\printbibliography`-Befehls zu jedem Segment, hat aber den zusätzlichen Vorteil vom automatischen Überspringen von Segmenten ohne Referenzen. Beachte, dass `\bibbysegment` für Referenzen in dem Segment 1 beginnt zu suchen. Er wird Referenzen, die außerhalb der `refsegment` Nachbarschaft angegeben werden, ignorieren, da sie zum Segment 0 zugeordnet sind. Für Anwendungsbeispiele siehe § 3.10.3. Die Optionen sind eine Teilmenge von denen, die `\printbibliography` unterstützt. Gültige Optionen sind `env`, `heading`, `prenote`, `postnote`, `maxnames`, `minnames`, `maxitems`, `minitems` und `section`.

`\bibbycategory` [*key=value, ...*]

Dieser Befehl führt automatisch Schleifen über alle Bibliographiekategorien aus. Dies ist äquivalent zu dem Übergeben eines `\printbibliography`-Befehls zu jeder Kategorie, hat aber den zusätzlichen Vorteil vom automatischen Überspringen von leeren Kategorien. Die Kategorien werden in der Reihenfolge bearbeitet, in der sie deklariert wurden. Für Anwendungsbeispiele siehe § 3.10.3. Die Optionen sind eine Teilmenge von denen, die `\printbibliography` unterstützt. Gültige Optionen sind `env`, `prenote`, `postnote`, `maxnames`, `minnames`, `maxitems`, `minitems` und `section`. Beachte, dass `heading` bei diesen Befehl nicht verfügbar ist. Der Name der aktuellen Kategorie wird automatisch als Titelname benutzt. Dies ist äquivalent, mit dem Übergeben von `heading=category` zu `\printbibliography` und besagt, dass es eine passende Titeldefinition für jede Kategorie geben muss.

`\printbibheading` [*key=value, ...*]

Dieser Befehl gibt einen Bibliographietitel aus, der mit `\defbibheading` definiert wurde. Er erfordert eine optionales Argument, das eine Liste von Optionen ist, die in

$\langle key \rangle = \langle value \rangle$ -Notation angegeben werden. Die Optionen sind eine kleine Teilmenge von denen, die `\printbibliography` unterstützt. Gültige Optionen sind `heading` und `title`. Automatisch benutzt dieser Befehl den Titel `bibliography`. Für Details siehe § 3.5.7. Für Anwendungsbeispiele siehe auch §§ 3.10.3 und 3.10.4.

### 3.5.3. Die Kürzelleiste

Wenn irgendein Eintrag ein `shorthand`-Feld beinhaltet, wird `biblatex` automatisch eine Liste von Kürzeln konstruieren, die zusätzlich zu der regulären Bibliographie hinzugefügt werden kann. Der folgenden Befehl gibt eine Liste von Kürzeln aus.

```
\printshorthands[$\langle key = value, \dots \rangle$]
```

Dieser Befehl gibt eine Liste von Kürzeln aus. Er erfordert ein optionales Argument, das eine Liste von Optionen ist, die in  $\langle key \rangle = \langle value \rangle$  angegeben werden. Gültige Optionen sind alle Optionen, die von `\printbibliography` (§ 3.5.2) unterstützt werden, außer `prefixnumbers`, `resetnumbers` und `omitnumbers`. Wenn es irgendwelche `refsection`-Nachbarschaften im Dokument gibt, wird die Liste von Kürzeln lokal zu diesen Nachbarschaften sein; für Details siehe § 3.5.4. Automatisch benutzt dieser Befehl die Titel `shorthands`. Für Details siehe § 3.5.7.

### 3.5.4. Bibliographiesektionen

Die `refsection`-Nachbarschaft wird im Dokumentkörper benutzt, um eine Referenzsektion zu markieren. Diese Nachbarschaft ist nützlich, wenn man separate, unabhängige Bibliographien und Listen von Kurzschriften in jedem Kapitel, den Sektionen oder in jedem anderen Teil des Dokuments haben will. Innerhalb einer Referenzsektion sind alle zitierte Arbeiten zugeteilte Labels, die lokal zu der Nachbarschaft sind. Eigentlich sind Referenzsektionen komplett unabhängig von den Dokumentteilen, wie `\chapter` und `\section`, obwohl sie am ehesten per Kapitel oder Sektion benutzt werden. Siehe die `refsection`-Paketoptionen in § 3.1.2.1 als einen Weg diese zu automatisieren. Für Anwendungsbeispiele siehe auch § 3.10.3.

```
\begin{refsection}[$\langle resource, \dots \rangle$]
\end{refsection}
```

Das optionale Argument ist eine Komma-geteilte-Liste von Ressourcen spezifisch für die Referenzsektion. Wenn das Argument fehlt, wird die Referenzsektion die vorgegebenen Ressourcelisten benutzen, wie sie mit `\addbibresource` in der Präambel festgelegt werden. Wenn das Argument bereitgestellt ist, ersetzt es die vorgegebene Ressourceliste. Allgemeine Ressourcen, die mit `\addglobalbib` festgelegt werden, werden immer betrachtet. `refsection`-Nachbarschaften dürfen nicht verschachtelt sein, aber man kann `refsegment`-Nachbarschaften innerhalb einer

`refsection` benutzen, um sie in Segmente aufzuteilen. Es benutzt die `section`-Option von `\printbibliography`, um eine Sektion auszuwählen, wenn die Bibliographie ausgegeben wird, und die dazugehörige Option von `\printshorthands`, wenn die Liste von Kürzeln ausgegeben wird. Bibliographiesektionen sind nummeriert, beginnend bei 1. Die Nummer der aktuellen Sektion wird auch in die Kopie-datei geschrieben. Alle Zitierungen, die außerhalb einer `refsection`-Nachbarschaft angegeben werden, sind der Sektion 0 zugeteilt. Wenn `\printbibliography` innerhalb einer `refsection` benutzt wird, wird sie automatisch die aktuelle Sektion auswählen. Die `section`-Option wird in diesen Fall nicht benötigt. Dies gilt auch für `\printshorthands`.

`\newrefsection` [*resource*, ...]

Dieser Befehl ist der `refsection`-Nachbarschaft ähnlich, außer das er eher ein selbstständiger Befehl ist als eine Nachbarschaft. Er beendet automatisch die bisherige Referenzsektion (gegebenfalls) und beginnt sofort eine neue. Beachten Sie, dass die Referenzsektion, die beim letzten `\newrefsection`-Befehl im Dokument startet, genau bis zum Ende des Dokuments erweitert wird. Man benutzt `\endrefsection`, wenn man es früher beenden will.

### 3.5.5. Bibliographiesegmente

Die `refsegment`-Nachbarschaft wird im Dokumentkörper benutzt, um ein Referenzsegment zu markieren. Diese Nachbarschaft ist nützlich, wenn man eine allgemeine Bibliographie will, die in Kapitel, Sektion oder jeden anderen Teil des Dokuments unterteilt wird. Eigentlich sind Referenzsegmente komplett unabhängig von den Dokumentteilen, wie etwa `\chapter` und `\section`, obwohl sie speziell per Kapitel oder Sektion benutzt werden. Siehe die `refsegment`-Paketoptionen in § 3.1.2.1 als einen Weg diese zu automatisieren. Für Anwendungsbeispiele siehe auch § 3.10.3.

`\begin{refsegment}`

`\end{refsegment}`

Der Unterschied zwischen einer `refsection`- und einer `refsegment`-Nachbarschaft ist, dass der erstere Labels erzeugt, die lokal zu der Nachbarschaft sind, während der letztere ein Ziel für den `segment`-Filter von `\printbibliography` unterstützt, ohne dass die Labels betroffen sind. Sie werden eindeutig über das ganze Dokument hinweg sein. `refsegment`-Nachbarschaften dürfen nicht verschachtelt sein, aber man kann sie in Verbindung mit `refsection` benutzen, um eine Referenzsektion in Segmente zu unterteilen. In diesem Fall sind die Segmente lokal zu der einschließenden `refsection`-Nachbarschaft. Benutze die `refsegment`-Option von `\printbibliography`, um ein Segment auszuwählen, wenn die Bibliographie ausgegeben wird. Die Referenzsegmente sind nummeriert, beginnend bei 1 und die

Nummer vom aktuellen Segment wird in die Kopiedatei geschrieben. Alle Zitierungen, die außerhalb einer `refsegment`-Nachbarschaft angegeben werden, sind der Sektion 0 zugeteilt. Im Gegensatz zu der `refsection`-Nachbarschaft wird das aktuelle Segment nicht automatisch ausgewählt, wenn `\printbibliography` innerhalb einer `refsegment`-Nachbarschaft benutzt wird.

`\newrefsegment` Dieser Befehl ist der `refsegment`-Nachbarschaft ähnlich, außer dass sie eher ein selbstständiger Befehl ist als eine Nachbarschaft. Er beendet automatisch das bisherige Referenzsegment (gegebenenfalls) und beginnt sofort ein neues. Beachte, dass das Referenzsegment, das beim letzten `\newrefsegment`-Befehl startet, bis zum Ende des Dokuments erweitert wird. Benutze `\endrefsegment` wenn man es früher beenden will.

### 3.5.6. Bibliographiekategorien

Bibliographiekategorien ermöglichen die Bibliographie in mehrere Teile, die verschiedenen Themen oder verschiedenen Typen von Referenzen gewidmet sind, zu teilen, zum Beispiel primäre und sekundäre Quellen. Für Anwendungsbeispiele siehe § 3.10.4.

`\DeclareBibliographyCategory{<category>}`

Deklariert eine neue `<category>`, die im Zusammenhang mit `\addtocategory` und den `category`- und `notcategory`-Filtern von `\printbibliography` benutzt wird. Dieser Befehl wird in der Präambel des Dokuments benutzt.

`\addtocategory{<category>}{<key>}`

Ordnet einen `<key>` einer `<category>` zu und wird in Verbindung mit den `category`- und `notcategory`-Filtern von `\printbibliography` verwendet. Dieser Befehl kann in die Präambel und in dem Dokumentgerüst geschrieben werden. Der `<key>` kann entweder ein einzelner Eingabeschlüssel sein oder eine durch Kommata getrennte Liste von Schlüsseln. Die Verknüpfung ist global.

### 3.5.7. Bibliografieüberschriften und Umgebungen

`\defbibenvironment{<name>}{<begin code>}{<end code>}{<item code>}`

Dieser Befehl definiert Bibliografie-Umgebungen. Der `<name>` ist ein Identifikator, der an die `env`-Option von `\printbibliography` und `\printshorthands` bei der Auswahl der Umgebung weitergegeben wird. Der `<begin code>` ist LaTeX-Code, welcher zu Beginn der Umgebung ausgeführt wird; der `<end code>` wird zum Schluß der Umgebung ausgeführt; der `<item code>` ist Code, der zu Beginn jeder Eingabe in die Bibliographie oder der Liste der Kurzschriften ausgeführt wird. Hier ein Beispiel einer Definition, basierend auf der Standard-`list`-Umgebung von LaTeX:

```

\defbibenvironment{bibliography}{
\list{}
{\setlength{\leftmargin}{\bibhang}%
\setlength{\itemindent}{-\leftmargin}%
\setlength{\itemsep}{\bibitemsep}%
\setlength{\parsep}{\bibparsep}} {
\endlist}
{\item}

```

Wie man in dem obigen Beispiel sehen kann, ist die Verwendung von `\defbibenvironment` annäherungsweise ähnlich zu `\newenvironment`, abgesehen davon, dass es hier ein zusätzliches vorgeschriebenes Argument für den *item code* gibt.

```

\defbibheading{<name>}[<title>]{<code>}

```

Dieser Befehl definiert Bibliografieüberschriften. Der *<name>* ist ein Identifikator, der an die `heading`-Option von `\printbibliography` oder `\printbibheading` und `\printshorthands` bei der Auswahl des Kopfes weitergegeben wird. Der *<code>* sollte LaTeX-Code sein, der, wenn gewünscht, eine vollständige Überschrift generiert, inklusive Seitenkopf und einem Eintrag in das Inhaltsverzeichnis. Wenn `\printbibliography` oder `\printshorthands` sich auf eine `title`-Option berufen, wird der Titel an die Kopfdefinition als #1 weitergegeben. Wenn nicht, wird anstelle dessen der Standardtitel, der durch das optionale *<title>*-Argument spezifiziert wird, als #1 weitergegeben. Typischerweise sind `\bibname`, `\refname` oder `\losname` die *<title>*-Argument (siehe § 4.9.2.1). Hier ein Beispiel einer einfachen Kopfdefinition:

```

\defbibheading{bibliography}[\bibname]{%
\chapter*{#1}% \markboth{#1}{#1}}

```

Die folgenden Überschriften, welche für die Verwendung mit `\printbibliography` und `\printbibheading` vorgesehen sind, sind vordefiniert:

### `bibliography`

Dies ist der Standardkopf, der von `\printbibliografie` verwendet wird, wenn die `heading`-Option nicht festgelegt ist. Seine Standarddefinition hängt von der Klasse des Dokuments ab (document class). Wenn die Klasse einen `\chapter`-Befehl bereitstellt, ist der Kopf ähnlich dem Bibliografiekopf der Standard-LaTeX-book-Klasse, das heißt, er verwendet `\chapter*`, um einen unnummerierten Kapitelkopf zu erzeugen, welcher nicht in das Inhaltsverzeichnis eingebunden wird. Wenn es keinen `\chapter`-Befehl gibt, ist er ähnlich dem Bibliografiekopf der Standard-LaTeX-article-Klasse, das heißt, er verwendet `\section*`, um einen unnummerierten

Teilabschnittskopf zu erzeugen, welcher nicht in das Inhaltsverzeichnis eingebunden wird. Die Zeichenfolge, welche im Kopf verwendet wird, hängt von der Dokumentklasse ab. In book-ähnlichen Klassen wird der Lokalisierungsstring `bibliography` verwendet, in anderen Klassen ist das `references` (siehe § 4.9.2). Für klassenspezifische Anleitungen siehe ebenfalls §§ 3.11.1 and 3.11.2.

#### `subbibliography`

Ist ähnlich wie `bibliography`, nur eine Sektionierungsebene darunter. Diese Kopfdefinition verwendet `\section*` anstelle von `\chapter*` in book-ähnlichen Klassen und ansonsten `\subsection*` anstelle von `\section*`.

#### `bibintoc`

Ähnlich wie `bibliography` oben, aber fügt einen Eintrag in das Inhaltsverzeichnis ein.

#### `subbibintoc`

Ähnlich wie `subbibliography` oben, aber fügt einen Eintrag in das Inhaltsverzeichnis ein.

#### `bibnumbered`

Ähnlich wie `bibliography` oben, aber verwendet `\chapter` oder `\section`, um einen nummerierten Kopf zu erzeugen, der ebenfalls in das Inhaltsverzeichnis eingefügt wird.

#### `subbibnumbered`

Ähnlich wie `subbibliography` oben, aber verwendet `\section` oder `\subsection`, um einen nummerierten Kopf zu erzeugen, der ebenfalls in das Inhaltsverzeichnis eingefügt wird.

#### `none`

Eine leere Kopfdefinition. Zu nehmen, um den Kopf abzustellen.

Die folgenden Kopfzeilen, bestimmt zur Verwendung mit `\printshorthands`, sind vordefiniert:

#### `shorthands`

Das ist der Standardkopf, der von `\printshorthands` ausgegeben wird, wenn die `optheading`-Option nicht gegeben ist. Sie ähnelt `bibliography` oben, mit dem Unterschied, dass sie die Lokalisierungsstring `shorthands` anstelle von `bibliography` oder `references` verwendet (siehe § 4.9.2). Für klassenspezifische Hinweise siehe auch §§ 3.11.1 and 3.11.2.

`losintoc`

Ähnlich wie `shorthands` oben, aber schreibt einen Eintrag in das Inhaltsverzeichnis.

`losnumbered`

Ähnlich zu `shorthands` oben, verwendet aber `\chapter` oder `\section`, um einen nummerierten Kopf zu erzeugen, der ebenfalls in das Inhaltsverzeichnis eingefügt wird.

### 3.5.8. Bibliografieanmerkungen

`\defbibnote`{*<name>*}{*<text>*}

Definiert die Bibliografienotiz *<name>*, die über die `prenote`- und `postnote`-Optionen von `\printbibliography` und `\printshorthands` verwendet wird. Der *<text>* kann jedes beliebige Stück Text sein, der möglicherweise auch mehrere Paragraphen umfasst und Satzatzdeklarationen enthält. Siehe ebenfalls § 3.11.6.

### 3.5.9. Bibliografiler und -kontrollen

`\defbibfilter`{*<name>*}{*<expression>*}

Definiert den üblichen Bibliografie-Filter *<name>*, der über die `filter`-Option von `\printbibliography` verwendet wird. Die *<expression>* ist ein komplexer Test, der auf den logischen Operatoren `and`, `or`, `not`, dem Gruppenseparator (`...`), und den folgenden Prüfoptionen besteht:

`segment`=*<integer>*

Gibt alle Einträge aus, die im Referenz-Segment *<integer>* eingetragen sind.

`type`=*<entrytype>*

Gibt alle Einträge aus, deren Eingabeart eine bestimmte *<entrytype>* ist.

`subtype`=*<subtype>*

Gibt alle Einträge aus, deren `entrysubtype` eine bestimmte *<subtype>* ist.

`keyword`=*<keyword>*

Gibt alle Einträge aus, deren `keywords`-Feld ein *<keyword>* beinhaltet. Wenn das *<keyword>* eine Leerstelle enthält, muss es in Klammern gesetzt werden.

`category`=*<category>*

Gibt alle Einträge mit `\addtcategory` aus, die einer *<category>* zugeordnet sind.

Hier ein Beispiel für einen Filter:

```
\defbibfilter{example}{%
(type=book or type=inbook) and keyword=abc and not keyword={x y z} }
```

Dieser Filter wird alle Einträge ausgeben, deren Eingabeart entweder @book oder @inbook ist und deren keywords-Feld das Schlüsselwort ,abc‘, jedoch nicht ,x y z‘ beinhaltet. Wie man im obigen Beispiel sehen kann, werden alle Elemente durch Leerstellen voneinander getrennt (Leerzeichen, Tabulatoren oder Zeilenenden). Es gibt keinen Zeichenabstand um gleichwertige Symbole herum. Die logischen Operatoren werden mit dem \ifboolexpr-Befehl aus dem etoolbox-Paket ausgewertet. Siehe etoolbox-Anleitung für Einzelheiten bezüglich der Syntax. Die Syntax des \ifthenelse-Befehls aus dem ifthen-Paket, welches in den älteren Versionen von biblatex Verwendung fand, wird nach wie vor unterstützt. Dies ist der selbe Test unter Verwendung von ifthen-ähnlicher Syntax:

```
\defbibfilter{example}{%
\(\ \type{book} \or \type{inbook} \) \and \keyword{abc}
\and \not \keyword{x y z} }
```

Man beachte, dass normale Filter lokal bezüglich des Referenzabschnitts sind, in welchem sie verwendet werden. Um einen anderen Abschnitt auszuwählen, benutzt man den section-Filter von \printbibliography. Das ist in einem normalen Filter nicht möglich.

```
\defbibcheck{<name>}{<code>}
```

Definiert den gebräuchlichen Bibliographiefilter *<name>*, der über die check-Option von \printbibliography verwendet wird. \defbibcheck ist vom Prinzip her ähnlich wie \defbibfilter, nur auf einer um einiges niedrigeren Ebene. Anders als ein Ausdruck auf hoher Ebene, ist der *<code>* LaTeX-Code, ähnlich dem Code, der in Treiberdefinitionen verwendet wird, welcher beliebige Tests durchführen kann, um zu entscheiden, ob ein Eintrag gedruckt werden soll oder nicht. Die Bibliographie-daten des betreffenden Eintrages sind vorhanden, wenn der *<code>* ausgeführt wird. Das Verwenden des Befehls \skipentry führt dazu, dass der momentane Eintrag übersprungen wird. Zum Beispiel: Der folgende Filter wird nur Einträge mit einem abstract-Feld ausgeben:

```
\defbibcheck{abstract}{%
\iffieldundef{abstract}{\skipentry}{} ...
\printbibliography[check=abstract]
```

Der folgende Test wird alle Einträge beinhalten, die vor dem Jahr 2000 veröffentlicht wurden:

```
\defbibcheck{recent}{% \iffieldint{year}
{\ifnumle"s{\thefield{year}}{2000} {\skipentry} {}} {\skipentry}}
```

Siehe Der Autorenführer, insbesondere §§ 4.6.2 and 4.6.3, für weitere Details.

### 3.5.10. Dynamische Mengeneinträge

Zusätzlich zu dem @set-Eingabetyp unterstützt biblatex auch dynamische Mengeneinträge, die auf einer Basis von ref-document/per-refsection definiert werden. Die folgenden Befehle definieren die Menge  $\langle key \rangle$  und können in der Präambel oder im Dokument verwendet werden:

```
\defbibentryset{\langle key \rangle}{\langle key1, key2, key3, ... \rangle}
```

Biber only

Der  $\langle key \rangle$  ist der Eingabeschlüssel der Menge, welcher wie jeder andere Eingabeschlüssel in Bezug auf die Menge verwendet wird. Der  $\langle key \rangle$  muss einzigartig sein und er darf nicht in Konflikt mit anderen Eingabeschlüsseln geraten. Das zweite Argument ist eine durch Kommata getrennte Liste von Eingabeschlüsseln, welche die Menge bilden. `\defbibentryset` beinhaltet ein Äquivalent eines `\nocite`-Befehls, das heißt, alle Mengen, welche deklariert wurden, werden auch in das Inhaltsverzeichnis geschrieben. Wenn eine Menge mehr als einmal deklariert wurde, wird nur der erste Aufruf von `\defbibentryset` die Menge bestimmen. Darauf folgende Definitionen des selben  $\langle key \rangle$  werden ignoriert und wie `\nocite\langle key \rangle` behandelt. Dynamische Mengeneinträge werden im Dokumentgerüst definiert und sind lokal bezüglich der umschließenden `refsection`-Umgebung, wenn vorhanden. Andernfalls werden sie der Sektion 0 des Literaturverzeichnisses zugeordnet. Diejenigen, die in der Präambel bestimmt wurden, ebenfalls. Man beachte, dass dynamische Mengeneinträge Biber benötigen. Sie werden mit keinem anderen Abschluss funktionieren. Siehe § 3.10.5 für weitere Details.

### 3.6. Zitierungsbefehle

Alle Befehle für Literaturangaben beinhalten generell ein zwingendes und zwei optionale Argumente. Die  $\langle prenote \rangle$  ist Text, der zu Beginn der Literaturstelle ausgegeben wird. Dies sind in der Regel Notizen wie ‚siehe‘ oder ‚vergleiche‘. Die  $\langle postnote \rangle$  ist Text, der zum Schluss der Literaturstelle ausgegeben wird. Das ist normalerweise eine Seitenzahl. Wenn man eine spezielle Vorbemerkung, aber keine Nachbemerkung machen möchte, muss man das zweite optionale Argument leer lassen, so wie bei `\cite[see] []{\key}`. Das  $\langle key \rangle$ -Argument für alle Befehle von Literaturstellen ist Pflicht. Dies ist ein Eingabeschlüssel oder eine durch Kommata getrennte Liste von Schlüsseln gemäss den Eingabeschlüsseln in der `bib`-Datei. Zusammengefasst haben alle unten aufgeführten grundlegenden Literaturbefehle im Wesentlichen folgende Syntax:

`\command`[*<prenote>*] [*<postnote>*]{*<keys>*}*<punctuation>*

Wenn die `autopunct`-Paket Option aus § 3.1.2.1 freigegeben ist, wird sie nach jeder *<punctuation>* suchen und sofort ihrem letzten Argument folgen. Das ist nützlich, um störende Satzzeichen nach der Literaturstelle zu vermeiden. Diese Dokumentation wurde mit `\DeclareAutoPunctuation` konfiguriert, siehe § 4.7.5 für Details.

### 3.6.1. Standardbefehle

Die folgenden Befehle werden durch den Stil für Literaturverweise definiert. Solche Stile können jede beliebige Anzahl spezieller Befehle enthalten, aber dies hier sind die Standardbefehle, die typischerweise Stile für die allgemeine Verwendung bereitstellen.

`\cite`[*<prenote>*] [*<postnote>*]{*<key>*}  
`\Cite`[*<prenote>*] [*<postnote>*]{*<key>*}

Dies sind die einfachen Befehle für Literaturangaben. Sie geben die Literaturstelle ohne jeden Zusatz aus, wie etwa Einschübe. Die numerischen und alphabetischen Formen schliessen das Label in eckige Klammern ein, da die Referenz andererseits doppeldeutig sein kann. `\Cite` ist genauso wie `\cite`, schreibt aber das Präfix des ersten Namens des Literaturverweises in Großbuchstaben, wenn die `useprefix`-Option vorhanden ist, vorausgesetzt, es gibt ein Namenspräfix und die Art der Literaturangabe gibt überhaupt alle Namen aus.

`\parencite`[*<prenote>*] [*<postnote>*]{*<key>*}  
`\Parencite`[*<prenote>*] [*<postnote>*]{*<key>*}

Diese Befehle verwenden ein Format, welches `\cite` gleicht, aber die Literaturangabe komplett in runden Klammern ausgibt. Die numerischen und alphabetischen Stile verwenden stattdessen eckige Klammern. `\Parencite` ist wie `\parencite`, schreibt aber das Präfix des ersten Namens des Literaturverweises in Großbuchstaben, wenn die `useprefix`-Option vorhanden ist, vorausgesetzt es gibt ein Namenspräfix.

`\footcite`[*<prenote>*] [*<postnote>*]{*<key>*}  
`\footcitetext`[*<prenote>*] [*<postnote>*]{*<key>*}

Dieser Befehl verwendet ein Format, welches ähnlich zu `\cite` ist, packt aber die komplette Quellenangabe in eine Fußnote und setzt am Ende einen Punkt. In der Fußnote wird das Präfix des Vornamen automatisch in Großbuchstaben geschrieben, wenn die `useprefix`-Option vorhanden ist, vorausgesetzt es gibt ein Namenspräfix und die Art der Literaturangabe gibt überhaupt alle Namen aus. `\footcitetext` unterscheidet sich von `\footcite` dadurch, dass es `\footnotetext` anstelle von `\footnote` verwendet.

### 3.6.2. Stilspezifische Befehle

Die folgenden zusätzlichen Befehle für Literaturangaben werden nur von einigen Arten von Quellenangaben bereitgestellt, welche mit diesem Paket geladen werden.

```
\textcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\Textcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Diese Befehle für Literaturangaben werden von allen nicht verbose Stildateien zur Verfügung gestellt, welche von diesem Paket geladen werden. Sie sind für die Verwendung im Textfluss vorgesehen, um das Subjekt des Satzes auszutauschen und geben den Autor oder Editor aus, gefolgt von den in runden Klammern stehenden anderen Kennzeichen der Literaturangabe. Abhängig von der Art der Literaturangabe kann das Kennzeichen eine Nummer, das Erscheinungsjahr, eine gekürzte Version des Titels oder Ähnliches sein. Die numerischen und alphabetischen Stile verwenden stattdessen eckige Klammern. `\Textcite` ist ähnlich wie `\textcite`, schreibt aber das Präfix des ersten Namens des Literaturverweises in Großbuchstaben, wenn die `useprefix`-Option vorhanden ist, vorausgesetzt es gibt ein Namenspräfix.

```
\smartcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\Smartcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Wie `\parencite` in einer Fußnote und wie `\footcite` im Dokument.

```
\cite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Dieser Befehl wird von allen Autor-Jahr- und Autor-Titel-Stilen bereitgestellt. Er ist ähnlich wie der normale `\cite`-Befehl, gibt aber lediglich jeweils Jahr oder Titel aus.

```
\parencite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Dieser Befehl wird von allen Autor-Jahr und Autor-Titel Stilen bereitgestellt. Er ist ähnlich wie der normale `\parencite`-Befehl, gibt aber lediglich jeweils Jahr oder Titel aus.

```
\supercite{⟨key⟩}
```

Dieser Befehl, welcher nur von den Numerik-Stilen unterstützt wird, gibt numerische Literaturangaben als Indizes ohne Klammern aus. Er verwendet `\supercitedelim` anstelle von `\multicitedelim` als Abgrenzung der Angaben. Man beachte, dass jedes `⟨prenote⟩`- und jedes `⟨postnote⟩`-Argument ignoriert wird. Sollten sie gegeben sein, wird `\supercite` sie streichen und eine Warnmeldung ausgeben.

### 3.6.3. Qualifizierte Listen von Literaturangaben

Dieses Paket unterstützt eine Klasse von speziellen Befehlen für Literaturangaben, genannt ‚multicite‘-Befehle. Das Besondere daran ist, dass ihre Argumente aus einer kompletten Liste von Literaturangaben bestehen, in denen jeder Eintrag einen vollständigen Literaturnachweis mit Vor- und/oder Schlussbemerkung darstellt. Dies ist besonders nützlich bei eingeschobenen Literaturangaben und solchen, die in Fußnoten geschrieben werden. Es ist ebenfalls möglich, eine Vor-/oder Schlussbemerkung in die vollständige Liste einzufügen. Diese Befehle für multiple Literaturangaben werden zu Beginn von hinten anstehenden Befehlen, so wie `\parencite` und `\footcite` eingebaut. Dieser Zitierstil unterstützt eine genaue Bestimmung für multiple Literaturangaben mit `\DeclareMultiCiteCommand` (siehe § 4.3.1). Das folgende Beispiel illustriert die Syntax solcher Befehle:

```
\parencites [35] {key1} [88--120] {key2} [23] {key3}
```

Das Format der Argumente ist ähnlich wie die regulären Befehle für Literaturangaben, nur dass lediglich ein Befehl gegeben ist. Wenn nur ein optionales Argument für die Beschreibung vorhanden ist, wird es als Schlussbemerkung ausgegeben. Möchte man eine spezifisch Vor-, jedoch keine Schlussbemerkung, muss man das zweite optionale Argument der betreffenden Beschreibung offen lassen:

```
\parencites [35] {key1} [chapter 2 in []] {key2} [23] {key3}
```

Zusätzlich dazu kann die komplette Liste auch eine Vor- und/oder eine Schlussbemerkung enthalten. Die Syntax dieser globalen Bemerkungen unterscheidet sich von anderen optionalen Argumenten darin, dass sie eher in geschweiften denn in üblichen Klammern gesetzt wird:

```
\parencites (and chapter 3) [35] {key1} [78] {key2} [23] {key3}
\parencites (Compare) () [35] {key1} [78] {key2} [23] {key3}
\parencites (See) (and the introduction) [35] {key1} [78] {key2} [23] {key3}
```

Man beachte, dass die Befehle für multiple Literaturangaben so lange nach Argumenten suchen, bis sie auf ein Merkmal stoßen, welches nicht der Beginn eines optionalen oder Pflichtarguments ist. Wenn einer linken geschweiften oder eckigen Klammer ein Befehl für eine multiple Literaturangabe folgt, muss man diese durch das Einfügen von `\relax` oder einem Backslash, gefolgt von ein Leerzeichen nach dem letzten gültigen Argument verschleiern. Das erzwingt das Ende des Suchvorgangs.

```
\parencites [35] {key1} [78] {key2} \relax [...]
\parencites [35] {key1} [78] {key2} _ { ... }
```

Dieses Paket unterstützt standardmäßig die folgenden Befehle, welche den regulären Befehlen von §§ 3.6.1 und 3.6.2 entsprechen:

```
\cites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
\Cites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
```

Ist die entsprechende Version multipler Literaturangaben von `\cite` und `\Cite`.

```
\parencites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
\Parencites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
```

Ist die entsprechende Version multipler Literaturangaben von `\parencite` und `\Parencite`.

```
\footcites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
\footcitetexts⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
```

Ist die entsprechende Version multipler Literaturangaben von `\footcite` und `\footcitetext`.

```
\smarcites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
\Smartcites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
```

Ist die entsprechende Version multipler Literaturangaben von `\smarcite` und `\Smartcite`.

```
\textcites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
\Textcites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
```

Ist die entsprechende Version multipler Literaturangaben von `\textcite` und `\Textcite`. Dieser Befehl wird nur von nicht „wortreichen“ Stilen unterstützt.

```
\supercites⟨pre⟩⟨post⟩ [⟨pre⟩ [⟨post⟩]{⟨key⟩}... [⟨pre⟩ [⟨post⟩]{⟨key⟩}
```

Ist die entsprechende Version multipler Literaturangaben von `\supercite`. Dieser Befehl wird lediglich von numerischen Stilen unterstützt.

#### 3.6.4. Stilunabhängige Befehle

Manchmal ist es nötig, Literaturangaben in der Quelldatei in einem Format anzugeben, welches nicht an einen speziellen Stil gebunden ist und in der Präambel global modifiziert werden kann. Das Format dieser Literaturangaben kann leicht durch das Laden eines anderen Stiles für Literaturangaben geändert werden. Dennoch, wenn man Befehle wie `\parencite` oder `\footcite` verwendet, ist die Art, wie die Literaturangaben in den Text eingebettet werden, trotzdem gewissermaßen vorprogrammiert. Die Idee hinter dem `\autocite`-Befehl ist die, die Heraufsetzung

übergeordneter Literaturangaben zu unterstützen, welche es möglich macht, inline-Literaturangaben global mit solchen zu tauschen, die als Fußnote (oder Index) angezeigt werden. Der `\autocite`-Befehl wird zu Beginn von hinten anstehenden Befehlen, so wie `\parencite` und `\footcite` eingebaut. Dieser Zitierstil unterstützt eine Definition von `\autocite` durch `\DeclareAutoCiteCommand` (siehe § 4.3.1) Diese Definition kann mithilfe der `\autocite`-Paketoption aus § 3.1.2.1 aktiviert werden. Der Stil der Literaturangabe von dieser Paketoption ist in einer Weise initialisiert, die für den Stil angemessen ist, siehe § 3.3.1 für Details. Man beachte, dass es gewisse Grenzen für die Heraufsetzung übergeordneter Literaturangaben gibt. Zum Beispiel integrieren inline-Literaturangaben die Autor-Jahr-Übersicht so eng mit dem Text, dass es virtuell unmöglich ist, diese in Fußnoten zu ändern. Der `\autocite`-Befehl ist nur verwendbar in Fällen, in denen man normalerweise `\parencite` oder `\footcite` (oder `\supercite` bei einem numerischen Stil) verwendet. Die Literaturangaben sollten am Ende des Satzes oder des Teilsatzes festgelegt werden, direkt vor dem letzten abschliessenden Satzzeichen und sie sollte nicht Teil des Satzes im grammatikalischen Sinne sein (wie `\textcite` zum Beispiel).

```
\autocite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\Autocite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Im Gegensatz zu anderen Befehlen für Literaturverzeichnisse sucht der `\autocite`-Befehl nicht nur nachfolgend nach Satzzeichen, die dem letzten Argument nachfolgen, um doppelte Satzzeichen zu vermeiden, tatsächlich verschiebt er sie sogar, wenn es erforderlich ist. Zum Beispiel wird bei `autocite=footnote` ein mitgeführtes Satzzeichen so verschoben, dass die Fußnotenmarkierung nach dem Zeichen ausgegeben wird. `\Autocite` ist so ähnlich wie `\autocite`, schreibt aber das Präfix des ersten Namens des Literaturverweises in Großbuchstaben, wenn die `useprefix`-Option vorhanden ist, vorausgesetzt es gibt ein Namenspräfix und die Art der Literaturangabe gibt überhaupt alle Namen aus.

```
\autocite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\Autocite*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Diese mit einem Stern versehene Variante von `\autocite` verhält sich nicht anders als die reguläre. Der Stern wird lediglich auf das Argument am Ende übergeben. Wenn zum Beispiel `\autocite` so konfiguriert werden soll, dass es `\parencite` verwendet, wird `\autocite* \parencite*` ausgeführt.

```
\autocites(⟨pre⟩)(⟨post⟩)[⟨pre⟩][⟨post⟩]{⟨key⟩}...[⟨pre⟩][⟨post⟩]{⟨key⟩}
\Autocites(⟨pre⟩)(⟨post⟩)[⟨pre⟩][⟨post⟩]{⟨key⟩}...[⟨pre⟩][⟨post⟩]{⟨key⟩}
```

Dies ist die Version für multiple Literaturangaben von `\autocite`. Sie findet und verschiebt ebenfalls Interpunktionen, wenn nötig. Man beachte, dass es keine Version

mit Stern gibt. `\Autocite` ist genau wie `\autocite`, schreibt aber das Präfix des ersten Namens des Literaturverweises in Großbuchstaben, wenn die `useprefix`-Option vorhanden ist, vorausgesetzt es gibt ein Namenspräfix und die Art der Literaturangabe gibt überhaupt alle Namen aus.

### 3.6.5. Textbefehle

Die folgenden Befehle werden vom Kern von `biblatex` bereitgestellt. Sie sind dafür ausgelegt, im fließenden Text verwendet zu werden. Man beachte, dass alle Textbefehle bei der Suche nach Literaturangaben ausgeschlossen sind.

```
\citeauthor[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\Citeauthor[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Diese Befehle geben die Autoren aus. Genauer gesagt, geben sie die `labelname`-Liste aus, die `author`, `editor` oder `translator` sein können. `\Citeauthor` ist wie `\citeauthor`, schreibt aber das Präfix des ersten Namens des Literaturverweises in Großbuchstaben, wenn die `useprefix`-Option vorhanden ist, vorausgesetzt es gibt ein Namenspräfix.

```
\citetitle[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\citetitle*[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Dieser Befehl gibt den Titel aus. Er verwendet den gekürzten Titel des Feldes `shorttitle`, wenn vorhanden. Andernfalls greift er auf das `title`-Feld mit dem vollen Titel zurück. Die Variante mit Stern gibt immer den vollen Titel aus.

```
\citeyear[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Dieser Befehl gibt das Jahr aus (`bibfieldyear`-Feld oder die Jahreskomponente von `date`).

```
\citedate[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Dieser Befehl gibt das komplette Datum aus (`date` oder `year`).

```
\citeurl[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Dieser Befehl gibt das `url`-Feld aus.

```
\parenttext{⟨text⟩}
```

Dieser Befehl setzt den `⟨text⟩` in kontextabhängige runde Klammern.

`\brackettext{⟨text⟩}`

Dieser Befehl setzt den `⟨text⟩` in kontextabhängige eckige Klammern.

### 3.6.6. Spezielle Befehle

Die folgenden speziellen Befehle werden ebenfalls vom vom `biblatex`-Kern unterstützt.

`\nocite{⟨key⟩}`

`\nocite{*}`

Dieser Befehl gleicht dem Standard-LaTeX-Befehl `\nocite`. Er fügt den `⟨key⟩` in die Bibliografie ein, ohne den Literaturverweis auszugeben. Ist der `⟨key⟩` ein Stern, werden alle verfügbaren Einträge in der `bib`-Datei in die Bibliografie eingefügt. Wie alle anderen Befehle für Literaturverweise sind `\nocite`-Befehle im Gerüst des Dokuments lokal zu der umgebenden `refsection`-Umgebung, sofern diese vorhanden ist. Im Unterschied zum Standard-LaTeX kann `\nocite` auch in der Präambel verwendet werden. In diesem Fall sind die Referenzen für die Sektion 0 bestimmt.

`\fullcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`

Dieser Befehl verwendet die Bibliographietreiber für den betreffenden Eingabetyp, um einen vollständigen Literaturverweis ähnlich zu dem Bibliographieeintrag zu erstellen. Deshalb ist er eher dem Bibliographiestil, denn dem Stil für Literaturverweise zuzuordnen.

`\footfullcite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}`

Ähneln `\fullcite`, setzt aber den kompletten Literaturverweis in eine Fußnote und setzt am Ende einen Punkt.

`\volcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}`

`\Volcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}`

Diese Befehle sind ähnlich wie `\cite` und `\Cite`, sie sind aber für Referenzen in mehrbändigen Arbeiten vorgesehen, welche mit der Nummer des Bandes und der Seitenzahl zitiert werden. Anstelle der `⟨postnote⟩` verwenden sie ein zwingendes `⟨volume⟩`- und ein optionales `⟨page⟩`-Argument. Seit sie lediglich eine Vorbemerkung bilden und diese an den `\cite` Befehl weitergeben, welcher vom Stil der Literaturangabe als `⟨postnote⟩`-Argument unterstützt wird, sind diese Befehle stilunabhängig.

`\pvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}`

`\Pvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}`

Gleicht `\volcite`, basiert aber auf `\parencite`.

```
\fvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\ftvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Gleicht `\volcite`, basiert aber auf `\footcite` beziehungsweise `\footcitetext`.

```
\svolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Svolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Gleicht `\volcite`, basiert aber auf `\smartcite`.

```
\tvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Tvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Gleicht `\volcite`, basiert aber auf `\textcite`.

```
\avolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Avolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Gleicht `\volcite`, basiert aber auf `\autocite`.

```
\notecite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\Notecite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Diese Befehle geben das `⟨prenote⟩`- und das `⟨postnote⟩`-Argument aus, jedoch nicht den Literaturverweis. Anstelle dessen wird ein `\nocite`-Befehl für jeden `⟨key⟩` ausgegeben. Das kann für Autoren nützlich sein, die implizite Literaturverweise in ihr Schriftstück mit einbeziehen, jedoch nur die Information dazu angeben und diese zuvor noch nicht im laufenden Text erwähnt haben, die aber trotzdem den Vorteil der automatischen `⟨postnote⟩`-Formatierung und der impliziten `\nocite`-Funktion ausnutzen wollen. Dies ist ein generischer, stilabhängiger Befehl für Literaturverweise. Spezielle Stile für Literaturangaben können elegantere Möglichkeiten für den selben Zweck bereitstellen. Die großgeschriebene Version erzwingt die Großschreibung (man beachte, dass dies nur dann verwendet werden kann, wenn die Bemerkung mit einem Befehl beginnt, der sensitiv ist für die `biblatex`-Zeichensuche).

```
\pnotecite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
\Pnotecite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Gleicht `\notecite`, doch die Bemerkungen werden in runden Klammern ausgegeben.

```
\fnotecite[⟨prenote⟩][⟨postnote⟩]{⟨key⟩}
```

Gleicht `\notecite`, doch die Bemerkungen werden in einer Fußnote ausgegeben.

### 3.6.7. Befehle niederer Ebenen

Die folgenden Befehle werden ebenfalls von dem biblatex-Kern unterstützt. Sie gewähren den Zugriff auf alle Listen und Felder auf einer niedrigeren Ebene.

`\citename`[*<prenote>*] [*<postnote>*]{*<key>*}[*<format>*]{*<name list>*}

Das *<format>* ist eine Formatierungsdirektive, die durch `\DeclareNameFormat` definiert wird. Formatierungsdirektiven werden in § 4.4.2 behandelt. Wenn das optionale Argument ausgelassen wird, fällt dieser Befehl zurück auf das Format `citename`. Das letzte Argument ist der Name des *<name list>*, dessen Sinn in § 2.2 erklärt wird.

`\citelist`[*<prenote>*] [*<postnote>*]{*<key>*}[*<format>*]{*<literal list>*}

Das *<format>* ist eine Formatierungsdirektive, die durch `\DeclareNameFormat` definiert wird. Formatierungsdirektiven werden in § 4.4.2 behandelt. Wenn das optionale Argument ausgelassen wird, fällt dieser Befehl zurück auf das Format `citelist`. Das letzte Argument ist der Name des *<literal list>*, dessen Sinn in § 2.2 erklärt wird.

`\citefield`[*<prenote>*] [*<postnote>*]{*<key>*}[*<format>*]{*<field>*}

Das *<format>* ist eine Formatierungsdirektive, die durch `\DeclareNameFormat` definiert wird. Formatierungsdirektiven werden in § 4.4.2 behandelt. Wenn das optionale Argument ausgelassen wird, fällt dieser Befehl zurück auf das Format `citefield`. Das letzte Argument ist der Name des *<field>*, dessen Sinn in § 2.2 erklärt wird.

### 3.6.8. Sonstige Befehle

Die Befehle in diesem Abschnitt sind kleine Helfer, die mit Literaturverweisen verwandt sind.

`\citereset` Dieser Befehl setzt den Stil des Literaturstelle zurück auf Null. Das kann nützlich sein, wenn der Stil sich wiederholende Literaturverweise durch Abkürzungen, wie *ibidem*, *idem*, *op. cit.* usw ersetzt und man eine vollständige Literaturangabe zu Beginn eines neuen Kapitels, Abschnitts oder einer anderen Position erzwingen möchte. Dieser Befehl führt einen spezifischen Initialisierungsaufhänger aus, der durch `\InitializeCitationStyle` aus § 4.3.1 definiert wird. Er setzt ebenfalls die eingebauten Literaturstellensucher dieses Paketes zurück. Der Reset wirkt sich aus auf die Tests `\ifciteseen`, `\ifentryseen`, `\ifciteibid` und `\ifciteidem`, welche in § 4.6.2 behandelt werden. Wenn der Befehl innerhalb einer `refsection`-Umgebung eingesetzt wird, ist die Rückstellung lokal zur momentanen `refsection`-Umgebung. Siehe ebenfalls beim `citereset`-Paket in § 3.1.2.1.

`\citereset*` Ähnlich zu `\citereset`, führt aber lediglich den Initialisierungsaufhänger des Stils aus, ohne die eingebauten Literaturstellensucher zurückzusetzen.

- `\mancite` Man verwendet diesen Befehl, um eingebettete Literaturstellen manuell zu markieren, wenn man automatisch generierte und manuell erstellte Literaturangaben gemischt verwenden will. Das ist besonders nützlich, wenn der Stil des Literaturverweises sich wiederholende Stellen durch Abkürzungen wie *ibidem* ersetzt, welche sonst nicht eindeutig oder irreführend sein können. `\mancite` sollte immer im gleichen Kontext wie die manuell erstellten Literaturverweise, zum Beispiel, wenn der Verweis als Fußnote gegeben ist, bezieht man `\mancite` in die Fußnote mit ein. Der `\mancite`-Befehl führt einen stilspezifischen Resetaufhänger aus, der mit dem `\OnManualCitation`-Befehl aus § 4.3.1 definiert wird. Er setzt ebenfalls die eingebetteten 'ibidem' und 'idem' Sucher des Paketes zurück. Der Reset wirkt sich auf `\ifciteibid`- und `\ifciteidem`-Tests aus, die in § 4.6.2 behandelt werden.
- `\pno` Dieser Befehl erzwingt ein einzelnes Seitenpräfix im *⟨postnote⟩*-Argument eines Befehls für Literaturstellen. Siehe § 3.11.3 für weitergehende Details und Verwendungsanleitungen. Man beachte, dass dieser Befehl nur lokal in Literaturstellen und Bibliografien vorhanden ist.
- `\ppno` Gleicht `\pno`, erzwingt aber ein abgegrenztes Präfix. Siehe § 3.11.3 für weitergehende Details und Verwendungsanleitungen. Man beachte, dass dieser Befehl nur lokal in Literaturstellen und Bibliografien vorhanden ist.
- `\nopp` Gleicht `\pno`, unterdrückt aber alle Präfixe. Siehe § 3.11.3 für weitergehende Details und Verwendungsanleitungen. Man beachte, dass dieser Befehl nur lokal in Literaturstellen und Bibliografien vorhanden ist.
- `\psq` Im Argument der *⟨postnote⟩* eines Befehls für Literaturangaben bezeichnet dieser Befehl einen Bereich von zwei Seiten, wo nur die Startseite gegeben ist. Siehe § 3.11.3 für weitere Details und Verwendungsanleitungen. Das ausgegebene Suffix ist der Lokalisierungsstring *sequens*, siehe § 4.9.2. Der eingebettete Abstand zwischen dem Suffix und der Seitenzahl kann durch die Neudefinition des Makros `\sqspace` modifiziert werden. Die Grundeinstellung ist eine unveränderbare Leerstelle innerhalb des Wortes. Man beachte, dass dieser Befehl nur lokal in Literaturstellen und Bibliografien vorhanden ist.
- `\psqq` Gleicht `\psq`, gibt aber einen Seitenabschnitt mit offenem Ende aus. Siehe § 3.11.3 für weitere Details und Anwendungsbeschreibungen. Das ausgegebene Suffix ist der Lokalisierungsstring *sequentes*, siehe § 4.9.2. Man beachte, dass dieser Befehl nur lokal in Literaturstellen und Bibliografien vorhanden ist.

#### `\RN{⟨integer⟩}`

Dieser Befehl gibt einen Ganzzahlwert als großgeschriebene römische Zahl aus. Die Formatierung des Zahlenzeichens kann durch die Neudefinition des Makros `\RNfont` modifiziert werden.

`\Rn{integer}`

Gleicht `\RN`, gibt aber eine kleingeschriebene römische Zahl aus. Die Formatierung des Zahlenzeichens kann durch die Neudefinition des Makros `\Rnfont` modifiziert werden.

### 3.6.9. natbib-kompatible Befehle

Das `natbib`-Paket lädt ein `natbib`-kompatibles Modul. Dieses Modul definiert Pseudonyme für die vom `natbib`-Paket unterstützten Befehle für Literaturverweise. Dies beinhaltet Parallelbezeichnungen für die Kernbefehle `\citet` und `\citep` ebenso, wie die Varianten `\citealt` und `\citealp`. Die Version mit Stern dieser Befehle, welche die komplette Autorenliste ausgibt, wird ebenfalls unterstützt. Der `\cite`-Befehl, welcher von `natbib` auf besondere Weise behandelt wird, wird nicht speziell behandelt. Die Textbefehle (`\citeauthor`, `\citeyear`, usw.) werden ebenfalls unterstützt, ebenso wie alle Befehle, welche das Namenspräfix in Großbuchstaben setzen (`\Citet`, `\Citep`, `\Citeauthor`, usw.). Die Das Setzen von Ersatznamen durch `\defcitealias`, `\citetalias`, und `\citepalias` ist möglich. Man beachte, dass die Kompatibilitätsbefehle nicht durch das Format der Literaturangaben des `natbib`-Pakets emuliert werden. Sie wandeln lediglich `natbib`-Befehle in Pseudonyme der funktional äquivalenten Möglichkeiten des `biblatex`-Paketes um. Das Format der Literaturverweise hängt vom Hauptstil der Literaturangaben ab. Trotzdem wird sich der Kompatibilitätsstil `\nameyear delim` angleichen, um mit dem Standardstil des `natbib`-Paketes übereinzustimmen.

### 3.6.10. mcite-artige Zitierungsbefehle

Biber only

Die `mcite`-Paketooption lädt ein spezielles Modul für Literaturangaben, welches `mcite`/`mciteplus`-ähnliche Befehle für Literaturverweise unterstützt. Genau genommen ist das, was die Module bereitstellen, die Hülle für Befehle des Hauptstils für Literaturstellen. Zum Beispiel der folgende Befehl:

```
\mcite{key1,setA,*keyA1,*keyA2,*keyA3,key2,setB,*keyB1,*keyB2,*keyB3}
```

Er ist tatsächlich äquivalent zu:

```
\defbibentryset{setA}{keyA1,keyA2,keyA3}%
\defbibentryset{setB}{keyB1,keyB2,keyB3}% \cite{key1,setA,key2,setB}
```

Der `\mcite`-Befehl arbeitet mit allen Stilen, seit es üblich ist, dass der letzte Befehl von `\cite` von vom Hauptstil für Literaturverweise kontrolliert wird. Das `mcite`-Modul stellt Hüllen für die Standardbefehle aus §§ 3.6.1 and 3.6.2 bereit. Siehe Tabelle 6 für einen Überblick. Vor- und Nachbemerkungen, ebenso wie die mit Stern versehene Version aller Befehle, werden unterstützt. Die Parameter werden an den hinten an stehenden Befehl weitergegeben. Zum Beispiel:

| Standardbefehl             | mcite-artiger Befehl        |
|----------------------------|-----------------------------|
| <code>\cite</code>         | <code>\mcite</code>         |
| <code>\Cite</code>         | <code>\Mcite</code>         |
| <code>\parencite</code>    | <code>\mparencite</code>    |
| <code>\Parencite</code>    | <code>\Mparencite</code>    |
| <code>\footcite</code>     | <code>\mfootcite</code>     |
| <code>\footcitetext</code> | <code>\mfootcitetext</code> |
| <code>\textcite</code>     | <code>\mtextcite</code>     |
| <code>\Textcite</code>     | <code>\Mtextcite</code>     |
| <code>\supercite</code>    | <code>\msupercite</code>    |

Tabelle 5: mcite-artige Befehle

```
\mcite*[pre] [post]{setA,*keyA1,*keyA2,*keyA3}
```

Er führt aus:

```
\defbibentryset{setA}{keyA1,keyA2,keyA3}%
\cite*[pre] [post]{setA}
```

Man beachte, dass das mcite-Modul kein Kompatibilitätsmodul ist. Es stellt Befehle zur Verfügung, die zwar sehr ähnlich, doch nicht gleich in Syntax und Funktion zum mcite-Befehl sind. Wechselt man von mcite/mciteplus zu biblatex, müssen die Altdateien (legacy files) aktualisiert werden. Mit mcite ist das erste Element der Gruppe von Literaturverweisen auch der Identifikator der Gruppe als Ganzes. Um ein Beispiel aus der mcite-Beschreibung zu übernehmen, es ist diese Gruppe:

```
\cite{glashow,*salam,*weinberg}
```

Sie besteht aus drei Einträgen und der Eingabeschlüssel des ersten fungiert auch als Identifikator der gesamten Gruppe. Im Gegensatz dazu ist eine biblatex-Eingabemenge eine eigenständige Einheit. Dafür ist ein eindeutig bestimmbarer Eingabeschlüssel notwendig, welcher der Menge zugewiesen wird, wie sie definiert ist:

```
\mcite{set1,*glashow,*salam,*weinberg}
```

Einmal definiert, kann die Eingabemenge wie jeder reguläre Eintrag in die bib-Datei behandelt werden. Verwendet man einen der numeric-Stile, welche mit biblatex geladen werden und die subentry-Option aktivieren, ist es auch möglich, auf Mengenelemente zu verweisen. Siehe Tabelle 6 für einige Beispiele. Eine erneute Angabe der originalen Definition ist nicht nötig, aber zulässig. Im Gegensatz zu mciteplus ist hier eine teilweise Neuangabe der ursprünglichen Definition unzulässig. Man verwendet stattdessen den Eingabeschlüssel der Menge.

| Eingabe                                             | Ausgabe | Kommentierung                                   |
|-----------------------------------------------------|---------|-------------------------------------------------|
| <code>\mcite{set1,*glashow,*salam,*weinberg}</code> | [1]     | Bestimmen und zitieren den Satz                 |
| <code>\mcite{set1}</code>                           | [1]     | Nachfolgendes Zitieren des Satzes               |
| <code>\cite{set1}</code>                            | [1]     | Übliches reguläres <code>\cite</code> -Arbeiten |
| <code>\mcite{set1,*glashow,*salam,*weinberg}</code> | [1]     | Redundant, aber permisibile                     |
| <code>\mcite{glashow}</code>                        | [1a]    | Zitieren ein Satzmitglied                       |
| <code>\cite{weinberg}</code>                        | [1c]    | Ebenso reguläres <code>\cite</code> -Arbeiten   |

Tabelle 6: `mcite`-artige Syntax (Beispiel der Ausgabe mit `style=numeric-` und `subentry`-Option)

### 3.7. Lokalisierungsbefehle

Das `biblatex`-Paket unterstützt Übersetzungen für Schlüsselbezeichnungen wie ‚`edition`‘ oder ‚`volume`‘ ebenso, wie Definitionen für sprachspezifische Merkmale, wie das Datumsformat und Ordinale. Diese Definitionen, welche automatisch geladen werden, können in der Präambel des Dokuments oder in der Konfigurationsdatei mit den in diesem Abschnitt vorgestellten Befehlen modifiziert oder erweitert werden.

```
\DefineBibliographyStrings{<language>}{<definitions>}
```

Dieser Befehl wird verwendet, um Lokalisierungsstrings zu definieren. Die `<language>` muss der Name einer Sprache aus dem `babel`-Paket sein, das heißt, einer der Identifikatoren aus Tabelle 1 auf Seite 30. Die `<definitions>` sind `<key>=<value>`-Paare, welche einem Ausdruck einen Identifikator zuordnen:

```
\DefineBibliographyStrings{american}{%
bibliography = {Bibliography},
shorthands = {Abbreviations},
editor = {editor},
editors = {editors},
}
```

Eine vollständige Liste aller Schlüssel, die von der Standardeinstellung bereitgestellt werden, findet sich in § 4.9.2. Man beachte, dass alle Ausdrücke in Großbuchstaben geschrieben sein sollten, was sie gewöhnlich auch sind, wenn sie innerhalb eines Satzes verwendet werden. Das `biblatex`-Paket schreibt das erste Wort automatisch groß, wenn das zu Beginn eines Satzes erforderlich ist. Ausdrücke, die für den Gebrauch in Überschriften vorgesehen sind, sollten auf eine Weise großgeschrieben werden, wie es für einen Titel angemessen ist. Im Gegensatz zu

`\DeclareBibliographyStrings` überschreibt `\DefineBibliographyStrings` sowohl die komplette, wie auch die abgekürzte Version des Strings. Siehe § 4.9.1 für weitere Details.

`\DefineBibliographyExtras{<language>}{<code>}`

Dieser Befehl wird verwendet, um sprachspezifische Merkmale wie das Datumsformat und Ordinale zu adaptieren. Die *<language>* muss der Name einer Sprache aus dem babel-Paket sein. Der *<code>*, der beliebiger LaTeX-Code sein kann, besteht normalerweise aus Neudefinitionen der Formatierungsbefehle aus § 3.8.2.

`\UndefineBibliographyExtras{<language>}{<code>}`

Dieser Befehl wird verwendet, um die Originaldefinition aller durch `\DefineBibliographyExtras` modifizierten Befehle wieder herzustellen. Wenn ein neudefinierter Befehl in § 3.8.2 inbegriffen ist, muss seine vorherige Definition nicht wieder hergestellt werden, seit diese Befehle sowieso in allen Sprachmodulen eingebaut wurden.

`\DefineHyphenationExceptions{<language>}{<text>}`

Dies ist ein LaTeX-Kopfende eines TeX-`\hyphenation`-Befehls, der Ausnahmen für die Silbentrennung definiert. Die *<language>* muss der Name einer Sprache aus dem babel-Paket sein. Der *<text>* ist eine durch Leerzeichen unterteilte Liste von Wörtern. Silbentrennungen werden durch einen Trennstrich markiert:

```
\DefineHyphenationExceptions{american}{%
hy-phen-ation ex-cep-tion }
```

`\NewBibliographyString{<key>}`

Dieser Befehl deklariert neue Lokalisierungsstrings, das heißt, er initialisiert einen neuen *<key>*, der in den *<definitions>* von `\DefineBibliographyStrings` verwendet wird. Das *<key>*-Argument kann ebenfalls eine durch Kommata getrennte Liste von Schlüsselnamen sein. Die Schlüssel, die in § 4.9.2 aufgelistet werden, sind durch die Standardeinstellung definiert.

### 3.8. Formatierungsbefehle

Die Befehle und Möglichkeiten, die in diesem Abschnitt erläutert werden, können zur Abstimmung der Formate von Literaturverweisen und Bibliographien verwendet werden.

#### 3.8.1. Generelle Befehle und Aufhänger

Die Befehle in diesem Abschnitt können mit `\renewcommand` in der Präambel neu definiert werden. Man beachte, dass alle Befehle, die mit `\mk...` beginnen, ein Argument benötigen. All diese Befehle sind in `biblatex.def` definiert.

- `\bibsetup` Beliebiger Kode, der zu Beginn der Bibliographie ausgeführt wird, für Befehle gedacht, die das Layout der Bibliographie betreffen.
- `\bibfont` Beliebiger Kode, der die in der Bibliographie verwendete Schriftart bestimmt. Dieser ist sehr ähnlich zu `\bibsetup`, ist aber für wechselnde Schriftarten gedacht.
- `\citesetup` Beliebiger Kode, der zu Beginn eines jeden Befehls für Literaturverweise ausgeführt wird.
- `\newblockpunct` Das Gliederungszeichen zwischen Bereichen unterschiedlicher Bedeutung, welches in § 4.7.1 erklärt wird. Die Standarddefinition wird von der Paketooption `block` kontrolliert (siehe § 3.1.2.1).
- `\newunitpunct` Das Gliederungszeichen zwischen Sinneinheiten, welches in § 4.7.1 erklärt wird. Das ist in der Regel ein Punkt oder ein Komma mit einem Leerzeichen. Die Standarddefinition ist ein Punkt mit einem Leerzeichen.
- `\finentrypunct` Die Interpunktion, die zum Schluss eines jeden Bibliographieeintrags gesetzt wird, normalerweise ein Punkt. Die Standarddefinition ist ein Punkt.
- `\bibnamedelima` Dieses Trennzeichen steuert den Abstand zwischen den Elementen, die ein Namens-  
teil sind. Es wird automatisch nach dem Vornamen eingefügt, wenn das Element  
länger als drei Zeichen lang ist und vor dem letzten Element ist. Die Standarddefini-  
tion ist ein wortinterner Raum, penaltisiert vom Wert des `highnamepenalty`-Zählers  
(§ 3.8.3). Sehen Sie bitte in § 3.11.4 für weitere Details. Biber only
- `\bibnamedelimb` Dieses Trennzeichen ist zwischen dem eingesetzten Element, das einen Namensteil  
bildet, der `\bibnamedelima` nicht verwendet. Die Standarddefinition ist ein wort-  
interner Raum, penaltisiert vom Wert des `lownamepenalty`-Zählers (§ 3.8.3).  
Sehen Sie bitte in § 3.11.4 für weitere Details. Biber only
- `\bibnamedelimc` Dieses Trennzeichen steuert den Abstand zwischen Namensteilen. Es ist eingefügt  
zwischen dem Namenspräfix und dem Nachnamen, wenn `useprefix=true`. Die  
Standarddefinition ist ein wortinterner Raum, penaltisiert vom Wert des  
`highnamepenalty`-Zählers (§ 3.8.3). Sehen Sie bitte in § 3.11.4 für weitere Details.
- `\bibnamedelimd` Dieses Trennzeichen ist zwischen allen Namensteilen, die `\bibnamedelimc` nicht  
verwendet. Die Standarddefinition ist ein wortinterner Raum, penaltisiert vom Wert  
des `lownamepenalty`-Zählers. (§ 3.8.3). Sehen Sie bitte in § 3.11.4 für weitere De-  
tails.
- `\bibnamedelimi` Dieses Trennzeichen ersetzt `\bibnamedelima/b` nach Initialen. Beachten Sie, dass  
dies nur für Initialen im `bib`-File gegeben ist, nicht für Initialen, die automatisch  
von `biblatex` generiert werden, die ihren eigenen Satz von Trennzeichen benutzen. Biber only

- `\bibinitperiod` Fügt die Interpunktion nach Initialen ein, es sei denn `\bibinithyphendelim` gilt. Die Standarddefinition ist ein Punkt (`\addot`). Sehen Sie bitte in § 3.11.4 für weitere Details. Biber only
- `\bibinitdelim` Der Abstand zwischen mehreren eingefügten Initialen, es sei denn `\bibinithyphendelim` gilt. Die Standarddefinition ist ein nicht zerbrechlicher Interwort-Raum. Sehen Sie bitte in § 3.11.4 für weitere Details. Biber only
- `\bibinithyphendelim` Die Satzzeichen zwischen den Initials von Namensteilen mit Bindestrich, ersetzt `\bibinitperiod` und `\bibinitdelim`. Die Standarddefinition ist ein Raum gefolgt von einem nichtzerbrechlichen Bindestrich. Sehen Sie bitte in § 3.11.4 für weitere Details. Biber only
- `\bibindexnamedelima` Ersetzt `\bibnamedelima` im Index.
- `\bibindexnamedelimb` Ersetzt `\bibnamedelimb` im Index.
- `\bibindexnamedelimc` Ersetzt `\bibnamedelimc` im Index.
- `\bibindexnamedelimd` Ersetzt `\bibnamedelimd` im Index.
- `\bibindexnamedelimi` Ersetzt `\bibnamedelimi` im Index.
- `\bibindexinitperiod` Ersetzt `\bibinitperiod` im Index.
- `\bibindexinitdelim` Ersetzt `\bibinitdelim` im Index.
- `\bibindexinithyphendelim` Ersetzt `\bibinithyphendelim` im Index.
- `\bibnamedash` Der Strich, der als Ersatz für wiederkehrende Autoren oder Herausgeber in der Bibliographie verwendet wird. Standardmäßig ist das ein ‚em‘ oder ‚en‘ Strich, abhängig von der Größe der Referenzliste.
- `\labelnamepunct` Das Gliederungszeichen, welches nach dem Namen ausgegeben wird, um die Bibliographie alphabetisch zu ordnen (author oder editor, falls das author-Feld nicht definiert ist). Bei den Standardstilen ersetzt dieses Trennzeichen an dieser Stelle `\newunitpunct`. Die Standarddefinition ist `\newunitpunct`, das heißt, es wird nicht anders verwendet als reguläre Abschnittsinterpunktionen.
- `\subtitlepunct` Das Gliederungszeichen, welches zwischen den Feldern title und subtitle, booktitle und booksubtitle, so wie maintitle und mainsubtitle gesetzt wird. Bei den Standardstilen ersetzt dieses Trennzeichen an dieser Stelle `\newunitpunct`. Die Standarddefinition ist `\newunitpunct`, das heißt, es wird nicht anders verwendet als reguläre Abschnittsinterpunktionen.

- `\intitlepunct` Das Gliederungszeichen zwischen dem Wort „in“ (aus) und dem darauffolgenden Titel in Anführungszeichen, so wie `@article`, `@inbook`, `@incollection`, usw. Die Standarddefinition ist ein Doppelpunkt mit einem Leerzeichen (zum Beispiel „Artikel, in: *Journal*“ oder „Titel, in: *Book*“). Man beachte, dass dies ein String ist und nicht nur eine Interpunktion. Wenn man keinen Doppelpunkt nach „aus“ haben möchte, fügt `\intitlepunct` trotzdem eine Leerstelle ein.
- `\bibpagespunct` Das Gliederungszeichen, welches vor dem `pages`-Feld eingefügt wird. Die Standardeinstellung ist ein Komma mit einem Leerzeichen.
- `\bibpagerefspunct` Das Gliederungszeichen, welches vor dem `pageref`-Feld ausgegeben wird. Die Standardeinstellung ist ein Leerzeichen.
- `\multinamedelim` Das Trennzeichen, welches zwischen multiplen Begriffen in einer Namensliste, wie `author` oder `editor`, wenn es mehr als zwei Namen in der Liste gibt. Die Standardeinstellung ist ein Komma, gefolgt von einem Leerzeichen. Siehe `\finalnamedelim` für ein Beispiel.<sup>21</sup>
- `\finalnamedelim` Das Trennzeichen, welches anstelle von `\multinamedelim` vor dem letzten Namen in einer Namensliste eingefügt wird. Die Standardeinstellung ist der Lokalisierungsterm `,and‘`, begrenzt durch Leerzeichen. Hier ein Beispiel:

```
Michel Goo"sens, Frank Mittelbach and
Alexander Samarin Edward Jones and Joe Williams
```

Das Komma im ersten Beispiel kommt von `\multinamedelim`, während der String `„and“` in beiden Beispielen von `\finalnamedelim` kommt. Siehen Sie ebenfalls `\finalandcomma` in § 3.8.2.

- `\revsdnamedelim` Ein zusätzliches Trennzeichen, welches nach dem Vornamen in einer Namensliste ausgegeben wird, wenn der Vorname gegeben ist. Die Standardvariante ist ein offener String, das bedeutet, es wird kein zusätzliches Trennzeichen ausgegeben. Hier ein Beispiel, welches eine Namensliste mit einem Komma als `\revsdnamedelim` zeigt:

```
Jones, Edward, and Joe Williams
```

In diesem Beispiel kommt das Komma nach `„Edward“` von `\revsdnamedelim`, während der String `„and“` von `\finalnamedelim` kommt, der zusätzlich zum vorherigen ausgegeben wird.

---

<sup>21</sup>Man beachte, dass `\multinamedelim` keinesfalls verwendet werden kann, wenn es nur zwei Namen in der Liste gibt. In diesem Fall verwenden die Standardstile `\finalnamedelim`.

- `\andothersdelim` Das Trennzeichen, welches vor dem Lokalisationsstring ‚and others‘ ausgegeben wird, wenn eine Namensliste wie `author` oder `editor` beschränkt wird. Die Standardversion ist ein Leerzeichen.
- `\multilistdelim` Das Trennzeichen, welches zwischen multiplen Einträgen in einer Wortliste, wie `publisher` oder `location`, wenn es mehr als zwei Einträge in dieser Liste gibt. Die Standardversion ist ein Komma mit einem Leerzeichen. Siehe `\multinamedelim` für weitere Erklärungen.
- `\finallistdelim` Das Trennzeichen, welches anstelle von `\multilistdelim` vor den letzten Eintrag in einer Wortliste gesetzt wird. Die Standardversion ist der Lokalisationsstring ‚and‘, der durch Leerzeichen abgegrenzt wird. Siehe `\finalnamedelim` für weitere Erklärungen.
- `\andmoredelim` Das Trennzeichen, welches vor dem Lokalisierungsstring ‚and more‘ ausgegeben wird, wenn eine Wortliste wie `publisher` oder `location` beschränkt wird. Die Standardversion ist ein Leerzeichen.
- `\multicitedelim` Das Trennzeichen, welches zwischen Literaturverweisen gesetzt wird, wenn multiple Eingabeschlüssel an einen einzelnen Befehl für Literaturangaben weitergegeben wird. Die Standardversion ist ein Semikolon mit einem Leerzeichen.
- `\supercitedelim` Gleicht `\multicitedelim`, wird aber nur vom `\supercite`-Befehl verwendet. Die Standardversion ist ein Komma.
- `\compcitedelim` Gleicht `\multicitedelim`, wird aber von mehreren Stilen für Literaturangaben zum ‚komprimieren‘ von multiplen Literaturverweisen verwendet.
- `\nametitledelim` Das Trennzeichen zwischen Autor/Herausgeber und dem Title aus Autor-Titel und einigen anderen „wortreichen“ Stilen für Literaturangaben. Die Standarddefinition ist ein Komma mit einem Leerzeichen.
- `\nameyear delim` Das Trennzeichen, welches zwischen Autor/Herausgeber und dem Jahr aus Autor-Jahr-Stilen für Literaturverweise gesetzt wird. Die Standardversion ist ein Leerzeichen.
- `\labelalphaothers` Ein String, der an den nicht numerischen Teil des `labelalpha`-Feldes angehängt wird (das heißt, der Feldanteil des Labels für Literaturverweise, welches von den alphabetischen Stilen für Literaturstellen verwendet wird), wenn die Abzahl von Autoren/Herausgebern `maxnames`-Grenze übersteigt oder die `author/editor`-Liste in der `bib`-Datei durch das Schlüsselwort ‚and others‘ beschränkt wurde. Das ist typischerweise eine einzelne Eigenschaft, wie ein Plussymbol oder ein Stern. Die Standardversion ist ein Plussymbol. Dieser Befehl kann auch durch einen leeren String neu definiert werden, um dieses Merkmal auszuschalten. In jedem Fall muss es in der Präambel neu definiert werden.

`\sortalphaothers` Gleicht `\labelalphaothers`, wird aber im Sortierprozess verwendet. Es in einem anderen Zusammenhang zu verwenden ist angebracht, wenn das Letztere einen Formatierungsbefehl enthält, zum Beispiel: Biber only

```
\renewcommand*\labelalphaothers{\textbf{+}}
\renewcommand*\sortalphaothers{+}
```

Wenn `\sortalphaothers` nicht neu definiert wird, wird er zu `\labelalphaothers`.

`\prenotedelim` Das Trennzeichen, welches nach dem `\prenote`-Argument eines Literaturstellen-Befehls ausgegeben wird. Siehe § 3.6 für Details. Die Voreinstellung ist ein Leerzeichen.

`\postnotedelim` Das Trennzeichen, welches nach dem `\postnote`-Argument eines Literaturstellen-Befehls ausgegeben wird. Siehe § 3.6 für Details. Die Voreinstellung ist ein Komma mit einem Leerzeichen.

`\mkbibnamelast`{*text*}

Dieser Befehl, welcher ein Argument benötigt, wird verwendet, um den Nachnamen aller Autoren, Herausgeber, Übersetzer usw. zu formatieren.

`\mkbibnamefirst`{*text*}

Gleicht `\mkbibnamelast`, ist aber für den Vornamen gedacht.

`\mkbibnameprefix`{*text*}

Gleicht `\mkbibnamelast`, ist aber für das Namenspräfix gedacht.

`\mkbibnameaffix`{*text*}

Gleicht `\mkbibnamelast`, ist aber für den Namenszusatz gedacht.

### 3.8.2. Sprachspezifische Befehle

Die Befehle in diesem Abschnitt sind sprachspezifisch. Wenn man sie neu definiert, muss man die neue Definition in einen `\DeclareBibliographyExtras`-Befehl einbetten, siehe § 3.7 für Details. Man beachte, dass alle Befehle, die mit `\mk...` beginnen, ein oder mehr Argumente benötigen.

`\bibrangedash` Der sprachspezifische Bindestrich, der für die Abgrenzung von Zahlen verwendet wird.

`\bibdatedash` Der sprachspezifische Bindestrich, der für die Abgrenzung von Datumsangaben verwendet wird.

- `\mkbibdatelong` Nimmt die Namen der drei Felder als Argumente, die sich auf die drei Datumskomponenten beziehen (in der Reihenfolge Jahr/Monat/Tag) und ihre Wirkung dafür verwenden, das Datum im sprachspezifischen ausgeschriebenen Format auszugeben.
- `\mkbibdateshort` Gleicht `\mkbibdatelong`, verwendet aber die kurze Version des sprachspezifischen Datumsformat.
- `\finalandcomma` Gibt die Kommata, die vor dem finalen ‚and‘ eingefügt wurden, in einer Liste aus, wenn möglich in der jeweiligen Sprache. Hier ein Beispiel:
- Michel Goo"sens, Frank Mittelbach, and Alexander Samarin
- `\finalandcomma` ist das Komma vor dem Wort ‚and‘. Siehe ebenfalls `\multinamedelim`, `\finalnamedelim` und `\revsdnamedelim` in § 3.8.1.

`\mkbibordinal`{*integer*}

Dieser Befehl, welcher einen ganzzahligen Wert als Argument beinhaltet, gibt eine normale Zahl aus.

`\mkbibmascord`{*integer*}

Gleicht `\mkbibordinal`, gibt aber eine männliche Ordnungszahl aus, wenn dies in der jeweiligen Sprache anwendbar ist.

`\mkbibfemord`{*integer*}

Gleicht `\mkbibordinal`, gibt aber eine weibliche Ordnungszahl aus, wenn dies in der jeweiligen Sprache anwendbar ist.

`\mkbibordedition`{*integer*}

Gleicht `\mkbibordinal`, ist aber für die Verwendung mit dem Begriff ‚Ausgabe‘ vorgesehen.

`\mkbibordseries`{*integer*}

Gleicht `\mkbibordinal`, ist aber für die Verwendung mit dem Begriff ‚Serie‘ vorgesehen.

### 3.8.3. Längen und Zähler

Die Längenregister und Zähler in diesem Abschnitt können gegebenenfalls in der Präambel mit `\setlength` und `\setcounter` geändert werden.

- `\bibhang` Die hängende Einrückung der Bibliografie, wenn angemessen. Diese Länge wird während der Ladezeit auf `\parindent` initialisiert.

- `\biblabelsep` Der Zeichenabstand zwischen Einträgen und ihren zugehörigen Labels in der Bibliographie. Es findet nur bei Bibliographiestilen Anwendung, die Kennsätze ausgeben, so wie die `numeric-` und `alphabetic-`Stile. Diese Länge wird durch die Wirkung von `\labelsep` während der Ladezeit doppelt initialisiert.
- `\bibitemsep` Der Zeilenabstand zwischen den einzelnen Einträgen in der Bibliographie. Diese Länge wird während der Ladezeit auf `\itemsep` initialisiert. Man beachte, dass `\bibitemsep`, `\bibnamesep` und `\bibinitsep` kumulativ sind. Sollten sie sich überschneiden, wird der Befehl mit der größten Wirkung ausgeführt.
- `\bibnamesep` Der Zeilenabstand, der zwischen zwei Einträgen der Bibliographie eingefügt wird, wann immer ein Eintrag mit einem Namen beginnt, der anders ist als der des vorherigen Eintrags. Die Voreinstellung ist null. Setzt man die Länge auf einen positiven Wert, der größer ist als `\bibitemsep`, wird die Bibliographie durch die Autoren-/Herausgebernamen gruppiert. Man beachte, dass `\bibitemsep`, `\bibnamesep` und `\bibinitsep` kumulativ sind. Sollten sie sich überschneiden, wird der Befehl mit der größten Wirkung ausgeführt.
- `\bibinitsep` Der Zeilenabstand, der zwischen zwei Einträgen der Bibliographie eingefügt wird, wann immer ein Eintrag mit einem Buchstaben beginnt, der anders ist als der des vorherigen Eintrags. Die Voreinstellung ist null. Setzt man die Länge auf einen positiven Wert, der größer ist als `\bibitemsep`, wird die Bibliographie alphabetisch gruppiert. Man beachte, dass `\bibitemsep`, `\bibnamesep` und `\bibinitsep` kumulativ sind. Sollten sie sich überschneiden, wird der Befehl mit der größten Wirkung ausgeführt.
- `\bibparsep` Der Zeilenabstand zwischen Paragraphen innerhalb eines Eintrags in der Bibliographie. Der Standardwert ist null.
- `abbrvpenalty` Dieser Zähler, der von Lokalisierungsmodulen verwendet wird, beinhaltet die Strafe, die in kurzen oder gekürzten Strings verwendet wird. Zum Beispiel ist ein Zeilenumbruch bei Ausdrücken wie „et al.“ oder „ed. by“ ungünstig, es sollte aber dennoch möglich sein, überfüllte Zellen zu vermeiden. Dieser Zähler wird während der Ladezeit auf `\hyphenpenalty` initialisiert. Diese Idee zwingt TeX dazu, den gesamten Ausdruck als ein einzelnes, durch Bindestrich trennbares Wort anzusehen, soweit es von einem Zeilenumbruch betroffen ist. Wenn man solche Zeilenumbrüche nicht mag, kann man einen höheren Wert verwenden. Wenn es einem egal ist, setzt man den Zähler auf null. Wenn man es unbedingt unterdrücken möchte, setzt man ihn auf ‚unendlich‘ (10 000 oder höher).<sup>22</sup>

<sup>22</sup>Die Standardwerte, die `abbrvpenalty`, `lownamepenalty` und `highnamepenalty` zugewiesen werden, sind absichtlich sehr niedrig, um überfüllte Zellen zu vermeiden. Das bedeutet, dass man kaum eine Auswirkung von Zeilenumbrüchen feststellen kann, wenn der Text im Blocksatz gesetzt ist. Wenn man diese Zähler auf 10 000 setzt, um eventuelle Trennstellen zu unterdrücken, wird man die Auswirkungen sehen können, kann aber auch mit überfüllten Zellen konfrontiert werden.

- `highnamepenalty` Dieser Zähler beinhaltet eine Strafe, welche den Zeilenumbruch bei Namen beeinflusst. In §§ 3.11.4 and 3.8.1 ist eine Erklärung zu finden. Dieser Zähler wird während der Ladezeit auf `\hyphenpenalty` initialisiert. Wenn man solche Zeilenumbrüche nicht mag, kann man einen höheren Wert verwenden. Wenn es einem egal ist, setzt man den Zähler auf null. Wenn man das traditionelle BibTeX-Verhalten bevorzugt (keine Zeilenumbrüche bei `highnamepenalty`-Trennstellen), setzt man ihn auf „unendlich“ (10 000 oder höher).
- `lownamepenalty` Gleicht `highnamepenalty`. Erklärungen finden sich in §§ 3.11.4 and 3.8.1. Dieser Zähler wird während der Ladezeit zur Hälfte auf `\hyphenpenalty` initialisiert. Wenn man solche Zeilenumbrüche nicht mag, kann man einen höheren Wert verwenden. Wenn es einem egal ist, setzt man den Zähler auf null.

### 3.8.4. Mehrzweckbefehle

Die Befehle dieses Abschnitts sind multifunktionale Befehle, die generell verfügbar sind, nicht nur für Zitierungen und die Bibliographie.

- `\bibellipsis` Ein elliptisches Symbol mit Klammern: `,[. . .]'`.
- `\noligature` Deaktiviert Ligaturen an dieser Stelle und fügt etwas Leerraum ein. Mit diesem Befehl können Standardligaturen wie `,fi'` and `,fl'` aufgebrochen werden. Es ist mit `|` vergleichbar, einer Kurzform, die vom `babel`-Paket für einige Sprachmodule zur Verfügung gestellt wird.
- `\hyphenate` Ein abhängiger Trennstrich. Im Gegensatz zum Standardbefehl `\-`, der die Silbentrennung für den Rest des Wortes ermöglicht. Es ist vergleichbar mit der Kurzform `"-`, die vom `babel`-Paket für einige Sprachmodule zur Verfügung gestellt wird.
- `\hyphen` Ein expliziter, zerbrechlicher Bindestrich in Komposita. Im Kontrast zu einem literalen `,-` ermöglicht dieser Befehl die Silbentrennung im Wortrest. Es ist vergleichbar mit der Kurzform `"=`, die vom `babel`-Paket für einige Sprachmodule zur Verfügung gestellt wird.
- `\nbhyphen` Ein expliziter, nicht-zerbrechlicher Bindestrich in Komposita. Im Kontrast zu einem literalen `,-` erlaubt dieser Befehl keine Zeilenumbrüche am Bindestrich, jedoch noch die Silbentrennung im Wortrest. Es ist vergleichbar mit der Kurzform `"~`, die vom `babel`-Paket für einige Sprachmodule zur Verfügung gestellt wird.

---

Man sollte bedenken, dass Zeilenumbrüche in der Bibliografie oft komplizierter als im Text sind und man nicht darauf zugreifen kann, einen Satz umzuformulieren. In manchen Fällen kann es wünschenswert sein, die gesamte Bibliographie in `\raggedright` zu setzen, um suboptimale Zeilenumbrüche zu vermeiden. In diesem Fall kann auch die kleinste Standardstrafe einen sichtbaren Unterschied zeigen.

`\nohyphenation` Ein allgemeiner Schalter, der die Silbentrennung lokal unterdrückt. Er sollte in der Regel auf eine Gruppe beschränkt werden.

`\textnohyphenation{⟨text⟩}`

Ähnlich zu `\nohyphenation`, jedoch beschränkt auf das `⟨text⟩`-Argument.

`\mknumalph{⟨integer⟩}`

Nimmt eine ganze Zahl aus dem Bereich 1–702 als Argument und wandelt es in einen String, wie die folgenden, um: 1=a, ..., 26=z, 27=aa, ..., 702=zz. Dies ist zur Verwendung für Formatierungsrichtlinien in den `extrayear`- und `extraalpha`-Feldern.

`\mkbibacro{⟨text⟩}`

Allgemeiner Befehl, der ein Acronym nimmt für die *small caps*-Variante der aktuellen Schrift, falls verfügbar, und wie anderweitig. Die Abkürzung sollte in Großbuchstaben eingegeben werden.

`\autocap{⟨character⟩}`

Automatische Konvertierung von `⟨character⟩` in seine Großform, wenn der Satzzeichen-Tracker von `biblatex` einen lokalen Groß-String nehmen würde. Dieser Befehl ist robust und nützlich für abhängige Großschreibung von bestimmten Zeichenketten in einem Eintrag. Beachten Sie, dass das `⟨character⟩`-Argument ein einzelnes Zeichen ist, das in Kleinbuchstaben einzugeben ist. Zum Beispiel:

```
\autocap{s}pecial issue
```

Es wird ‚Special issue‘ oder ‚special issue‘, als Passendes. Wenn die Zeichenfolge mit Großschreibung startet, wird ein flektiertes Zeichen in Ascii-Notation ausgegeben, einschließlich den zugehörigen Akzentbefehl für das `⟨character⟩`-Argument, wie nachfolgend:

```
\autocap{\oe}uvres
```

Daraus ergibt sich ‚Œuvres‘ or ‚œuvres‘.

### 3.9. Sprachspezifische Anmerkungen

Die Möglichkeiten, die in diesem Kapitel diskutiert werden, sind spezifisch für bestimmte Localization-Module.

### 3.9.1. Amerikanisches Englisch

Das amerikanische Lokalisation-Modul verwendet `\uspunctuation` aus § 4.7.5, um eine ‚American-style‘-Interpunktion zu ermöglichen. Wenn diese Funktion aktiviert ist, werden alle Kommata und Punkte innerhalb von Anführungszeichen nach `\mkbibquote` verschoben. Wenn Sie diese Funktion deaktivieren möchten, verwenden Sie `\stdpunctuation` wie folgt:

```
\DefineBibliographyExtras{american}{%
\stdpunctuation }
```

Standardmäßig ist die ‚American punctuation‘-Funktion durch das `american` Lokalisation-Modul aktiviert. Der obige Code ist nur erforderlich, wenn Sie die amerikanische Lokalisierung ohne die amerikanische Zeichensetzung möchten. Ansonsten würde es redundant zu anderen Sprachen sein.

Es ist äußerst ratsam, zu spezifizieren `american`, `british`, `australian`, etc., anstatt nur `english`, wenn Sie das `babel`-Paket laden, um etwaige Verwechslungen zu vermeiden. Ältere Versionen des `babel`-Pakets verwenden die Option `english` als Alias für die Option `british`; die neueren behandeln es als Alias für `american`. Das `biblatex`-Paket behandelt im Wesentlichen `english` als Alias für `american`, außer bei der obigen Funktion, die nur aktiviert wird, wenn `american` angefordert wird.

### 3.9.2. Spanisch

Die Handhabung von dem Wort ‚und‘ ist im Spanischen komplizierter als in anderen Sprachen, welche von diesem Paket unterstützt werden, weil es ein ‚y‘ oder ein ‚e‘ sein kann, je nachdem wie das nächste Wort beginnt. Hierfür nutzt das spanische Lokalisierungsmodul nicht den Lokalisierungsstring ‚and‘ sondern ein spezielles internes ‚smart and‘ Kommando. Das Verhalten dieses Kommandos wird von dem `smartand` Zähler kontrolliert.

`smartand` Dieser Zähler kontrolliert das Verhalten des internen ‚smart and‘ Kommandos. Wenn er auf 1 steht, druckt er ‚y‘ oder ‚e‘, vom Kontext abhängig. Wenn er auf 2 steht, druckt er immer ein ‚y‘. Wenn er auf 3 steht, druckt er immer ein ‚e‘. Wenn er auf 0 steht, dann ist die Funktion ‚smart and‘ ausgeschaltet. Dieser Zähler wird beim Start mit 0 initialisiert und kann in der Präambel geändert werden. Man beachte, dass die Änderung des Zählers auf einen positiven Wert impliziert, dass das Spanische Lokalisierungsmodul `\finalnamedelim` und `\finallistdelim` ignoriert.

`\forceE` Man benutzt dieses Kommando in `bib` Dateien, falls `biblatex` das ‚und‘ vor einem bestimmten Wort falsch druckt. Wie der Name schon sagt, wird ein ‚e‘ erzwungen. Dieses Kommando muss in einer bestimmten Weise benutzt werden, um BibTeX nicht zu verwirren. Hier ist ein Beispiel:

```
author = {Edward Jones and Eoin Maguire},
author = {Edward Jones and {\forceE{E}}oin Maguire},
```

Man beachte, dass der erste Buchstabe der betreffenden Wortkomponente durch ein Argument für `\forceE` gegeben ist und das gesamte Konstrukt in ein zusätzliches Paar geschweifte Klammern gefasst ist.

`\forceY` Analog zu `\forceE`, erzwingt aber ‚y‘

### 3.9.3. Griechisch

Das griechische Lokalisierungsmodul benötigt den UTF-8 Support. Es arbeitet mit keiner anderen Kodierung. Im Allgemeinen ist das `biblatex`-Paket kompatibel mit dem `inputenc`-Paket und XeLaTeX. Das `ucs`-Paket funktioniert nicht. Nachdem das `inputenc` Standard-`utf8`-Modul keine Bildzeichen Funktion für das Griechische bereitstellt, bleibt für griechische Nutzer nur noch XeLaTeX. Man beachte, dass man eventuell zusätzliche Pakete für griechische Schriften laden muss. Als Daumenregel gilt, eine Einstellung, welche für normale griechische Dokumente funktioniert, sollte auch mit `biblatex` funktionieren. Allerdings gibt es eine fundamentale Einschränkung. Zum jetzigen Zeitpunkt unterstützt `biblatex` nicht das Wechseln zwischen Schriften. Griechische Titel in der Bibliographie sollten gut funktionieren, vorausgesetzt man nutzt Biber als Backend, aber englische und andere Titel in der Bibliographie könnten als griechische Buchstaben dargestellt werden. Falls man Bibliographien mit mehreren Schriften braucht, ist XeLaTeX die einzige vernünftige Wahl.

## 3.10. Anwendungshinweise

Der folgende Abschnitt gibt einen grundlegenden Überblick über das `biblatex` Paket und zeigt einige typische Anwendungsszenarien.

### 3.10.1. Überblick

Die Benutzung des `biblatex` Pakets ist etwas anders als die Benutzung traditioneller BibTeX-Stile und ähnlicher Pakete. Bevor wir zu einigen spezifischen Anwendungsproblemen kommen, schauen wir uns dafür zunächst die Struktur eines typischen Dokuments an:

```
\documentclass{...}
\usepackage[...]{biblatex}
\addbibresource{bibfile.bib}
\begin{document} \cite{...} ...
\printbibliography
\end{document}
```

In dem traditionellen BibTeX dient das `\bibliography` Kommando zwei Zwecken. Es markiert den Ort der Bibliographie und spezifiziert den/die `bib` Datei(en). Die Dateierweiterung wird weggelassen. Mit `biblatex` werden Quellen in der Präambel mit `\addbibresource` durch die Nutzung des vollen Namens mit `.bib` Endung spezifiziert. Die Bibliographie wird mit dem `\printbibliography` Kommando gedruckt, welches eventuell mehrere Male benutzt wird (siehe § 3.5 für Details). Der Dokumentkörper kann mehrere Zitierkommandos enthalten (§ 3.6). Die Verarbeitung dieses Beispiels erfordert eine bestimmte Prozedur, die nachfolgend gezeigt wird. Vorausgesetzt unser Beispiel heißt `example.tex` und unsere Bibliographiedaten sind in `bibfile.bib`, dann ist die Prozedur folgende:

1. Laufenlassen `latex` von `example.tex`. Falls die Datei irgendwelche Zitate enthält, verlangt `biblatex` die betreffenden Daten von BibTeX, indem es Kommandos in die Hilfsdatei `example.aux` schreibt.
2. Laufenlassen `bibtex` von `example.aux`. BibTeX wird die Daten von `bibfile.bib` empfangen und schreibt sie in die Hilfsdatei `example.bbl` in einem Format, welches von `biblatex` verarbeitet werden kann.
3. Laufenlassen `latex` von `example.tex`. `biblatex` liest die Daten aus `example.bbl` ein und druckt alle Zitate ebenfalls in die Bibliographie.

Immer wenn ein Verweis auf eine Arbeit, welche bisher nicht zitiert worden ist, hinzugefügt wird, muss diese Prozedur wiederholt werden. Dies kann auch der Fall sein, wenn der letzte Verweis auf eine Arbeit, welche zuvor zitiert worden ist, entfernt wurde, weil in diesem Fall sich einige Zitatbeschriftungen geändert haben können. Im Gegensatz zum traditionellen BibTeX ist es normalerweise nicht nötig, zweimal `latex` nach `bibtex` laufen zu lassen, solange die Behandlung der bibliographischen Daten betroffen ist.<sup>23</sup> Man beachte, dass dies nur auf die einfachsten Fälle zutrifft. Die Nutzung des `xref` Feldes oder des `entryset` Feldes kann einen zusätzlichen LaTeX/BibTeX/LaTeX Zyklus voraussetzen. Einige andere Hilfsmittel, welche durch `biblatex` bereit gestellt werden, können ebenso einen zusätzlichen `latex`-Lauf benötigen, um bestimmte Verweise und Seitentracking richtig zu erhalten. In diesem Fall können die üblichen Warnnachrichten wie „There were undefined references“ and „Label(s) may have changed. Rerun to get cross-references right“ angezeigt werden. Wenn man Biber als Backend benutzt, ist der Arbeitsablauf der gleiche.

### 3.10.2. Hilfsdateien

Laut Default-Setting nutzt das `biblatex`-Paket nur die Haupt-`aux`-Datei. Auch, wenn Zitierkommandos in einer Datei via `\include` eingefügt sind, welches seine eige-

<sup>23</sup>Das ist der Fall, sofern die `defernumbers`-Paketooption aktiviert ist. Diese Option erfordert zwei LaTeX Läufe, nachdem die Daten in die `bbl` Datei durch BibTeX exportiert worden.

ne aux-Datei besitzt, musst man nur BibTeX mit der Haupt-aux-Datei laufen lassen. Falls man `refsection`-Umgebungen in einem Dokument (vgl. § 3.10.3) nutzt, wird `biblatex` eine zusätzliche aux-Datei für jede `refsection`-Umgebung erstellen. In diesem Fall muss man auch `bibtex` mit jeder zusätzlichen aux-Datei laufen lassen. Der Name der zusätzlichen aux-Datei besteht aus dem Basisnamen der Haupt-Input-Datei mit dem Anhang `-blx` und einer fortlaufenden Nummer am Ende. Das `biblatex`-Paket gibt eine Warnung aus, die alle Dateien auflistet, welche einen zusätzlichen BibTeX Lauf benötigen. Mit dem einfachen Beispiel, dargestellt in § 3.10.1, würde es folgende Warnung ausgeben:

```
Package biblatex Warning: Please (re)run BibTeX
on the file(s): (biblatex) example.aux (biblatex)
and rerun LaTeX afterwards.
```

Falls die Eingabedatei drei `refsection`-Umgebungen enthält, würde die Warnung folgenderweise aussehen:

```
Package biblatex Warning: Please (re)run BibTeX
on the file(s): (biblatex)
example1-blx.aux (biblatex)
example2-blx.aux (biblatex)
example3-blx.aux (biblatex)
and rerun LaTeX afterwards.
```

Abgesehen von diesen aux-Dateien, nutzt `biblatex` eine zusätzliche `bib`-Datei mit der gleichen Endung, um bestimmte Kontrollparameter an BibTeX übergeben zu können. In dem Beispiel oben würde diese Datei `example-blx.bib` heißen. In Fall eines Namenskonfliktes kann man die Endung ändern, indem man das Makro `\blxauxsuffic` in der Dokumentpräambel neu definiert. Wenn man Biber benutzt, schreibt `biblatex` eine Kontrolldatei namens `example.bcf` und ignoriert `\blxauxsuffix`. Es gibt in diesem Fall dann auch eine `bib` Hilfsdatei.

Man sollte beachten, dass `biblatex` keine Dateien überschreibt, die es nicht selbst erstellt hat. Alle Hilfsdateien, die automatisch durch dieses Paket erzeugt worden, beginnen mit einer speziellen Signaturzeile. Bevor eine Datei überschrieben wird (ausgenommen die aux-Hauptdatei, welche von LaTeX geführt wird), untersucht `biblatex` die erste Zeile der Datei, um sicher zu gehen, dass kein Namenskonflikt besteht. Falls die entsprechende Datei keine Signaturzeile besitzt, gibt `biblatex` sofort eine Fehlernachricht aus und bricht ab, bevor eine Ausgabe erzeugt wurde. In diesem Fall sollte man alle störenden Dateien löschen, welche unbeabsichtigt im Arbeitsverzeichnis verblieben waren. Falls der Fehler weiterhin besteht, kann ein Namenskonflikt mit einer Datei, die in einer der TeX Installationsbaumes erzeugt worden ist, bestehen. Da die Installationsbäume normalerweise keine aux-Dateien enthalten und die Zeichenkette `-blx` recht exotisch in dem Namen einer `bib`-Datei ist, ist es recht unwahrscheinlich, aber theoretisch möglich. Falls man herausfindet,

dass dies doch tatsächlich der Fall ist, sollte man `\blxauxsuffix` dauerhaft in der `biblatex` Konfigurationsdatei, `biblatex.cfg` umdefinieren.

### 3.10.3. Mehrere Bibliographien

In einer Sammlung von Artikeln von verschiedenen Autoren, wie ein Konferenzprotokoll zum Beispiel, ist es üblicher eine Bibliographie für jeden Artikel zu haben, statt eine globale für das ganze Buch. Im Beispiel unten wird jeder Artikel durch ein separates `\chapter` mit seiner eigenen Bibliographie dargestellt. Man beachte, dass `biblatex` eine zusätzliche `aux`-Datei für jede `refsection`-Umgebung anlegt. Diese Dateien müssen von BibTeX ebenfalls verarbeitet werden, siehe § 3.10.2 für Details.

```
\documentclass{...}
\usepackage{biblatex}
\addbibresource{...}
\begin{document}
\chapter{...}
\begin{refsection} ...
\printbibliography[heading=subbibliography]
\end{refsection}
\chapter{...}
\begin{refsection} ...
\printbibliography[heading=subbibliography]
\end{refsection}
\end{document}
```

Falls `\printbibliography` in einer `refsection`-Umgebung benutzt wird, beschränkt es automatisch den Bereich der Referenzliste auf die betreffende `refsection` Umgebung. Für eine wachsende Bibliographie, welche in Kapitel unterteilt ist, jedoch am Ende des Buches gedruckt wird, nutzt man die `section`-Option von `\printbibliography`, um einen Referenzabschnitt zu wählen, wie man im nächsten Beispiel sieht.

```
\documentclass{...}
\usepackage{biblatex}
\defbibheading{subbibliography}{%
\section*{References for Chapter
\ref{refsection:\therefsection}}}
\addbibresource{...} \begin{document}
\chapter{...}
\begin{refsection} ... \end{refsection}
\chapter{...}
\begin{refsection} ... \end{refsection}
\printbibheading
\printbibliography[section=1,heading=subbibliography]
```

```
\printbibliography[section=2,heading=subbibliography]
\end{document}
```

Man beachte im Beispiel oben die Definition des Bibliographietitels. Das ist die Definition, welche auf die Untertitel in der Bibliographie aufpasst. Der Haupttitel wird in diesem Fall durch ein einfaches `\chapter` Kommando erzeugt. Das `biblatex`-Paket setzt automatisch eine Beschriftung an den Anfang einer jeden `refsection`-Umgebung, indem es das Standard-`\label`-Kommando nutzt. Die Bezeichnung, die genutzt wird, ist die Zeichenkette `refsection:` gefolgt von der Nummer der betreffenden `refsection` Umgebung. Die Nummer des aktuellen Abschnitts ist durch den `refsection` Zähler verfügbar. Wenn man die `section` Option von `\printbibliography` nutzt, ist der Zähler auch lokal gesetzt. Das heißt, dass man den Zähler für Titeldefinitionen nutzen kann, um Untertitel, wie „Referenz für Kapitel 3“, wie oben gezeigt, zu drucken. Man kann außerdem den Titel des betreffenden Kapitels als Untertitel nutzen indem man das `nameref`-Paket lädt und `\nameref` statt `\ref` nutzt:

```
\usepackage{nameref}
\defbibheading{subbibliography}{%
\section*{\nameref{refsection:\therefsection}}}
```

Da ein `\printbibliography` Kommando für jeden Teil der unterteilten Bibliographie lästig ist, bietet `biblatex` ein Kürzel. Das `\bibbysection` Kommando durchläuft automatisch alle Referenzabschnitte. Das ist das äquivalent für ein `\printbibliography`-Kommando für jeden Abschnitt, jedoch hat es den zusätzlichen Vorteil, des automatischen Überspringens von Abschnitten ohne Referenzen. In dem Beispiel oben würde die Bibliographie dann folgendermaßen erzeugt werden:

```
\printbibheading
\bibbysection[heading=subbibliography]
```

Wenn man ein Format ohne eine wachsende Bibliographie nutzt, welche in Kapitel (oder irgendeine andere Dokumentunterteilung) gegliedert ist, ist es vermutlich angebrachter `refsegment`- statt `refsection`-Umgebungen zu nutzen. Der Unterschied ist, dass die `refsection` Umgebung lokal für die Umgebung die Beschriftungen erzeugt, während `refsegment` die Beschriftungserzeugung nicht beeinflusst, daher sind sie somit einzigartig über das gesamte Dokument. Man beachte, dass `refsegment`-Umgebungen keine zusätzlichen `aux`-Dateien benötigen. Das nächste Beispiel könnte ebenso in § 3.10.4 genutzt werden, da es optisch eine globale Bibliographie, in mehrere Segmente unterteilt, erzeugt.

```
\documentclass{...}
\usepackage{biblatex}
```

```

\defbibheading{subbibliography}{%
\section*{References for Chapter \ref{refsegment:\therefsegment}}}
\addbibresource{...}
\begin{document}
\chapter{...}
\begin{refsegment} ... \end{refsegment}
\chapter{...}
\begin{refsegment} ... \end{refsegment}
\printbibheading
\printbibliography[segment=1,heading=subbibliography]
\printbibliography[segment=2,heading=subbibliography]
\end{document}

```

Die Nutzung von `refsegment` ist ähnlich zu `refsection` und es gibt ebenso eine entsprechende `segment`-Option für `\printbibliography`. Das `biblatex`-Paket setzt automatisch eine Bezeichnung an den Beginn von jeder `refsegment`-Umgebung, indem es die Zeichenkette `refsegment:` gefolgt von der Nummer der entsprechenden `refsegment`-Umgebung als Bezeichner nutzt. Es gibt einen angepassten `refsegment`-Zähler, welcher für die Titeldefinitionen, wie oben gezeigt, genutzt werden kann. Wie bei den Referenzabschnitten gibt es auch hier ein Kurzkommando, welches automatisch über alle Referenzsegment läuft:

```

\printbibheading
\bibbysegment[heading=subbibliography]

```

Das ist äquivalent zu je einem `\printbibliography` Kommando für jedes Segment.

### 3.10.4. Unterteilte Bibliographien

Es ist sehr üblich eine Bibliographie nach bestimmten Kriterien zu unterteilen. Zum Beispiel: Man will vielleicht gedruckte und online Quellen separat drucken oder eine Bibliographie in Primär- und Sekundärliteratur teilen. Der erstere Fall ist einfach, weil man den Eintragstypen als Kriterium für die `type`- und `notttype`-Filter von `\printbibliography` nutzen kann. Das nächste Beispiel demonstriert außerdem, wie man angepasste Untertitel für zwei Teile der Bibliographie erzeugt.

```

\documentclass{...}
\usepackage{biblatex}
\addbibresource{...}
\begin{document}
...
\printbibheading
\printbibliography[notttype=online,heading=subbibliography,
title={Printed Sources}]

```

```
\printbibliography[type=online,heading=subbibliography,
 title={Online Sources}]
\end{document}
```

Man kann auch mehr als zwei Unterteilungen nutzen:

```
\printbibliography[type=article,...]
\printbibliography[type=book,...]
\printbibliography[notype=article,notype=book,...]
```

Es ist sogar möglich, eine Kette von verschiedenen Filtertypen zu geben:

```
\printbibliography[section=2,type=book,keyword=abc,notkeyword=xyz]
```

Das würde alle zitierten Arbeiten in Referenzabschnitt 2 drucken, dessen Eintragstyp @book ist und dessen keywords-Feld das Schlüsselwort ‚abc‘ aber nicht ‚xyz‘ beinhaltet. Wenn man Bibliographiefilter in Verbindung mit einem numerischen Stil verwendet, siehe § 3.11.5. Falls man einen komplexen Filter mit bedingten Ausdrücken benötigt, nutzt man die `filter`-Option in Verbindung mit einem spezifizierten Filter, definiert durch `\defbibfilter`. Siehe § 3.5.9 für Details über spezifizierte Filter.

```
\documentclass{...}
\usepackage{biblatex}
\addbibresource{...}
\begin{document}
...
\printbibheading
\printbibliography[keyword=primary,heading=subbibliography,%
 title={Primary Sources}]
\printbibliography[keyword=secondary,heading=subbibliography,%
 title={Secondary Sources}]
\end{document}
```

Eine Unterteilung der Bibliographie in Primär- und Sekundärliteratur ist mit einem keyword-Filter möglich, wie im Beispiel oben gezeigt. In diesem Fall, mit nur zwei Unterbereichen, wäre es ausreichend, ein Schlüsselwort als Filterkriterium zu nutzen:

```
\printbibliography[keyword=primary,...]
\printbibliography[notkeyword=primary,...]
```

Da biblatex keine Möglichkeit hat, zu wissen, ob ein Eintrag in der Bibliographie Primär- oder Sekundärliteratur sein soll, muss man die Bibliographiefilter mit den benötigten Daten unterstützen, indem man ein keywords-Feld für jeden Eintrag in die bib Datei hinzufügt. Diese Schlüsselwörter können dann als Ziele für

den keyword- und nokeyword-Filter benutzt werden, wie oben gezeigt. Es kann eine gute Idee sein, solche Schlüsselwörter während des Aufbaus einer bib Datei hinzuzufügen.

```
@Book{key,
keywords = {primary,some,other,keywords},
...
}
```

Ein alternativer Weg, um die Referenzliste zu unterteilen, sind Bibliographiekategorien. Sie unterscheiden sich von dem schlüsselwortbasierten Denkansatz, der im Beispiel oben gezeigt ist, darin, dass sie auf dem Dokumentlevel arbeiten und keine Änderungen in der bib-Datei benötigen.

```
\documentclass{...}
\usepackage{biblatex}
\DeclareBibliographyCategory{primary}
\DeclareBibliographyCategory{secondary}
\addtocategory{primary}{key1,key3,key6}
\addtocategory{secondary}{key2,key4,key5}
\addbibresource{...}
\begin{document}
...
\printbibheading
\printbibliography[category=primary,heading=subbibliography,%
title={Primary Sources}]
\printbibliography[category=secondary,heading=subbibliography,%
title={Secondary Sources}]
\end{document}
```

In diesem Fall wäre es ausreichend, nur eine Kategorie zu nutzen:

```
\printbibliography[category=primary,...]
\printbibliography[notcategory=primary,...]
```

Es ist eine gute Idee, alle benutzten Kategorien in der Bibliographie explizit zu deklarieren, weil es ein `\bibbycategory`-Kommando gibt, welches automatisch über alle Kategorien läuft. Es ist äquivalent zu je einem `\printbibliographie`-Kommando für jede Kategorie, in der Reihenfolge, in der sie deklariert wurden.

```
\documentclass{...}
\usepackage{biblatex}
\DeclareBibliographyCategory{primary}
\DeclareBibliographyCategory{secondary}
\addtocategory{primary}{key1,key3,key6}
\addtocategory{secondary}{key2,key4,key5}
```

```

\defbibheading{primary}{\section*{Primary Sources}}
\defbibheading{secondary}{\section*{Secondary Sources}}
\addbibresource{...}
\begin{document}
...
\printbibheading \bibbycategory
\end{document}

```

Die Behandlung der Titel ist in diesen Fall unterschiedlich von `\bibbysection` und `\bibbysegment`. `\bibbycategory` nutzt den Namen der aktuellen Kategorie als Titelname: Das ist äquivalent zur Weitergabe von `heading=\langle category \rangle` zu `\printbibliography` und impliziert, dass man einen angepassten Titel für jede Kategorie braucht.

### 3.10.5. Eintragssätze

Ein Eintragssatz ist eine Gruppe von Einträgen, die als eine Einzelreferenz zitiert wurden und als einzelne Einheit in der Bibliographie gelistet sind. Das `biblatex`-Paket unterstützt zwei Typen von Eintragssätzen. Statische Eintragssätze sind in der `bib`-Datei definiert, wie jeder andere Eintrag. Dynamische Eintragssätze sind mit `\defbibentryset` (§ 3.5.10) auf einer *per-document/per-refsection* Basis in der Dokumentpräambel oder im Dokumentenkörper definiert. Dieser Abschnitt beschäftigt sich mit der Definition von Eintragssätzen; Stilautoren sollten sich auch § 4.11.1 für mehr Informationen ansehen.

**3.10.5.1. Statische Eintragssätze** Statische Eintragssätze sind in der `bib`-Datei definiert, wie jeder andere Eintrag. Wenn man als Backend Biber benutzt, definiert man einen Eintragssatz einfach, indem man einen Eintrag vom Typ `@set` hinzufügt. Der Eintrag hat ein `entryset`-Feld, welches die Mitglieder des Satzes als durch Kommas geteilte Liste von Eintragungsschlüsseln definiert:

Biber only

```

@Set{set1,
entryset = {key1,key2,key3},
}

```

Einträge können Teil von einem Satz in einem Dokument/Refsektion und alleinstehende Referenzen in einem anderen sein, je nachdem, ob es den `@set`-Eintrag gibt oder nicht. Falls der `@set`-Eintrag zitiert wird, werden die Satzmitglieder automatisch gruppiert. Falls nicht, werden sie wie jeder andere reguläre Eintrag gehandhabt.

Wenn man BibTeX benutzt, welches keine native Unterstützung für Eintragssätze besitzt, macht es mehr Arbeit einen Eintragssatz anzulegen. BibTeX benötigt `entryset`- und `crossref`-Felder, um in einer besonderen Art benutzt zu werden.

Die Mitglieder von einem Satz sind in dem `entryset`-Feld des `@set`-Eintrags gegeben. Der `@set`-Eintrag benötigt außerdem ein `crossref`-Feld, welches auf den ersten Schlüssel im `entryset`-Feld zeigt. Zusätzlich benötigen alle Mitglieder des Satzes `entryset`-Felder, welche umgekehrte Zeiger auf den Eintragungsschlüssel vom `@set` Kopfeintrags sind:

```
@Set{set1,
entryset = {key1,key2,key3},
crossref = {key1}, } @Article{key1,
entryset = {set1},
author = {...},
title = {...},
... }
@InCollection{key2,
entryset = {set1},
author = {...},
title = {...},
... }
@Article{key3,
entryset = {set1},
author = {...},
title = {...},
... }
```

Man beachte, dass das Zitieren von einem Satzmitglied automatisch den ganzen Satz mit BibTeX laden wird. Falls man sich auf eine Einheit als Teil eines Satzes in einem Dokument/Refektion und als allein stehende Referenz in einem anderen beziehen will, braucht man zwei individuelle Einträge mit BibTeX.

**3.10.5.2. Dynamische Eintragungssätze** Dynamische Eintragungssätze werden ähnlich gebildet wie statische und funktionieren auch ähnlich. Der Hauptunterschied ist, dass sie in der Dokumentpräambel oder während des Schreibens im Dokumentenkörper durch Nutzung des `\defbibentryset`-Kommandos aus § 3.5.10 definiert werden:

Biber only

```
\defbibentryset{set1}{key1,key2,key3}
```

Dynamische Eintragungssätze im Dokumentenkörper sind lokal bezüglich der umgebenden `refsection`-Umgebung, falls es eine gibt. Anderenfalls zählen sie zum Referenzabschnitt 0. Die, die in der Präambel definiert wurden, gehören zum Referenzabschnitt 0. Man beachte, dass dynamische Eintragungssätze Biber benötigen. Sie arbeiten mit keinem anderen Backend.

### 3.10.6. Elektronische Herausgabeinformation

Das biblatex-Paket unterstützt drei Felder für elektronische Herausgabeinformationen: `eprint`, `eprinttype` und `eprintclass`. Das `eprint`-Feld ist ein wortwörtliches Feld, analog zu `doi`, welches den Bezeichner einer Einheit enthält. Das `eprinttype`-Feld beinhaltet den Quellennamen, z. B. den Namen von der Seite oder des elektronischen Archivs. Das optionale `eprintclass`-Feld ist für zusätzliche Informationen speziell zu der Quelle, die durch das `eprinttype`-Feld angegeben wird, vorgesehen. Das könnte ein Abschnitt, ein Pfad, Klassifikationsinformationen, etc sein. Falls ein `eprinttype`-Feld verfügbar ist, wird es den Standardstil als einen wortgetreuen Bezeichner benutzen. In dem folgenden Beispiel würde „Resource: identifier“ statt dem allgemeinen „eprint: identifier“ gedruckt werden:

```
eprint = {identifier},
eprinttype = {Resource},
```

Die Standardstile besitzen einen geeigneten Support für einige Onlinearchive. Für arXiv Referenzen setze man den Bezeichner in das `eprint`-Feld und die Zeichenkette `arxiv` in das `eprinttype`-Feld:

```
eprint = {math/0307200v3},
eprinttype = {arxiv},
```

Für Dokumente, welche nach dem neuen Bezeichnerschema (April 2007 und später) benutzt werden, fügt man eine einfache Klassifikation in das `eprintclass`-Feld hinzu:

```
eprint = {1008.2849v1},
eprinttype = {arxiv},
eprintclass = {cs.DS},
```

Es gibt zwei Aliase, welche die Integration von arXiv-Einträgen erleichtern. `archiveprefix` wird als Alias für `eprinttype` behandelt; `primaryclass` ist ein Pseudonym für `eprintclass`. Falls Hyperlinks aktiviert sind, wird der Eprint-Bezeichner in einen Link zu `arxiv.org` transformiert. Man betrachte für weitere Details die Paketoptionen `arxiv` in § 3.1.2.1.

Für JSTOR-Referenzen schreibe man die feste JSTOR-Nummer in das `eprint`-Feld und die Zeichenkette `jstor` in das `eprinttype`-Feld:

```
eprint = {number},
eprinttype = {jstor},
```

Wenn man das Exportfeature von JSTOR benutzt, um Zitate in das BibTeX-Format zu exportieren, dann nutzt JSTOR das `url`-Feld standardmäßig (wo die `number` einzigartig ist und ein fester Bezeichner ist):

```
url = {http://www.jstor.org/stable/number},
```

Dies wird, wie erwartet, funktionieren, jedoch tendieren vollständige URLs dazu, die Bibliographie zu verstopfen. Mit den eprint-Feldern werden die Standardstile das besser lesbare „JSTOR: *number*“ Format nutzen, welches auch Hyperlinks unterstützt. Die *number* wird ein anklickbarer Link, falls der hyperref-Support aktiviert ist.

Für PubMed-Referenzen schreibe man den festen PubMed-Bezeichner in das eprint-Feld und die Zeichenkette pubmed in das eprinttype-Feld. Das sieht dann folgendermaßen aus:

```
url = {http://www.ncbi.nlm.nih.gov/pubmed/pmid},
```

Es wird:

```
eprint = {pmid},
eprinttype = {pubmed},
```

Und die Standardstile drucken „PMID: *pmid*“ statt der langen URL. Falls der hyperref-Support aktiviert ist, wird *pmid* ein anklickbarer Link zu PubMed werden.

Für Handles (HDLs) schreibe man den Handle in das eprint-Feld und die Zeichenkette hdl in das eprinttype-Feld:

```
eprint = {handle},
eprinttype = {hdl},
```

Für Google Books Referenzen schreibe man den Bezeichner von Google in das eprint-Feld und den String googlebooks in das eprinttype-Feld. Dies sieht zum Beispiel folgendermaßen aus:

```
url = {http://books.google.com/books?id=XXu4AkRVBBoC},
```

Es würde zu Folgendem werden:

```
eprint = {XXu4AkRVBBoC},
eprinttype = {googlebooks},
```

Und die Standardstile würden „Google Books: XXu4AkRVBBoC“ statt der vollen URL drucken. Falls der hyperref-Support aktiviert ist, wird der Bezeichner zu einem anklickbaren Link zu Google Books.<sup>24</sup>

Man beachte, dass eprint ein wörtliches Feld ist. Es gibt immer den Bezeichner in der unmodifizierten Form. Zum Beispiel ist es unnötig ein \_ durch ein \\_ zu ersetzen. Siehe dazu auch § 4.11.2, um zu sehen, wie man einen geeigneten Support für andere Eprint Quellen hinzufügt.

<sup>24</sup>Man beachte, dass die Google Books id eher ein ‚interner‘ Wert zu sein scheint. Zum jetzigen Zeitpunkt scheint es keinen Weg zu geben, um nach einer id in Google Books zu suchen. Man könnte die Nutzung von url in diesem Fall bevorzugen.

### 3.10.7. Externe Inhaltsangaben und Anmerkungen

Stile, die die Felder `abstract` und/oder `annotation` ausgeben, können einen alternativen Weg unterstützen, um Inhaltsangaben oder Anmerkungen zu der Bibliographie hinzuzufügen. Statt den Text in der `bib`-Datei einzubinden, kann er ebenso in einer externen LaTeX-Datei gespeichert werden. Zum Beispiel: Statt in der `bib`-Datei zu sagen

```
@Article{key1,
...
abstract = {This is an abstract of entry 'key1'.}
}
```

kann man eine Datei namens `bibabstract-key1.tex` erzeugen und die Inhaltsangabe in diese Datei zu schreiben:

```
Dies ist eine Inhaltsangabe zu dem Eintrag 'key1'.
\endinput
```

Der Name der externen Datei muss vor dem Eintragungsschlüssel das Präfix `bibabstract-` beziehungsweise `bibannotation-` haben. Man kann diese Präfixe ändern, indem man `\bibabstractprefix` und `\bibannotationsprefix` umdefiniert. Man beachte, dass dieses Feature explizit aktiviert werden muss, indem man die Paketoption `loadfiles` aus § 3.1.2.1 setzt. Die Option ist standardmäßig aus Performancegründen deaktiviert. Man beachte außerdem, dass jedes `abstract-` und `annotation-`Feld in der `bib`-Datei den Vozug vor externen Dateien besitzt. Die Nutzung von externen Dateien wird dringend empfohlen, falls man lange Inhaltsangaben oder viele Anmerkungen hat, da dies den Speicherbedarf signifikant erhöht. Es ist außerdem angenehmer einen Text in einer zugehörigen LaTeX-Datei zu editieren. Stilautoren sollten sich § 4.11.3 für weitere Informationen anschauen.

## 3.11. Hinweise und Warnungen

Dieser Abschnitt bietet zusätzliche Nutzungshinweise und zeigt einige übliche Probleme und potentielle Missverständnisse.

### 3.11.1. Nutzung mit KOMA-Script-Klassen

Wenn man `liblatex` in Verbindung mit der `scrbook-`, `scrreprt-` oder `scartcl-`Klasse benutzt, sind die Titel `bibliography` und `shorthands` von § 3.5.7 ansprechbar für die bibliographiebezogenen Optionen dieser Klassen.<sup>25</sup> Man kann die Stan-

<sup>25</sup>Dies bezieht sich auf die traditionelle Syntax dieser Optionen (`bibtoc` und `bibtocnumbered`) und ebenso auf die `<key>=<value>`-Syntax, die in KOMA-Script 3.x hinzugefügt wurde, z. B. auf

ardtittel überschreiben, indem man die `heading`-Option von `\printbibliography`, `\printbibheading` und `\printshorthands` benutzt. Siehe §§ 3.5.2, 3.5.3, 3.5.7 für Details. Alle Standardtitel werden in der Ladezeit derart adaptiert, dass sie sich mit dem Verhalten der Klassen vermischen. Falls eine der oberen Klassen entdeckt wird, stellt `biblatex` folgende zusätzliche Tests bereit, welche nützlich für nutzerspezifische Titeldefinitionen sein können:

`\ifkomabibtoc{<true>}{<false>}`

Wird zu `<true>`, falls die Klasse die Bibliographie zu dem Inhaltsverzeichnis hinzufügen würde und zu `<false>` anderenfalls.

`\ifkomabibtocnumbered{<true>}{<false>}`

Wird zu `<true>`, falls die Klasse die Bibliographie zu dem Inhaltsverzeichnis als nummerierten Abschnitt hinzufügen würde, anderenfalls wird es zu `<false>`. Falls dieser Test `<true>` bringt, wird `\ifkomabibtoc` ebenso immer `<true>` zurückgeben, aber nicht anderes herum.

### 3.11.2. Nutzung mit der Memoir-Klasse

Wenn man `biblatex` mit der `memoir`-Klasse benutzt, haben die meisten Klassenhilfen für die Adaption der Bibliographie keinen Effekt. Man benutze stattdessen die dementsprechenden Hilfen von dem Paket (§§ 3.5.2, 3.5.7, 3.5.8). Statt `\bibsection` von `memoir` umzudefinieren, nutze man die `heading`-Option von `\printbibliography` und `\defbibheading` (§§ 3.5.2 and 3.5.7). Statt `\prebibhook` und `\postbibhook` nutze man die `prenote`- und `postnote`-Option von `\printbibliography` und `\defbibnote` (§§ 3.5.2 and 3.5.8). Alle Standardtitel werden beim Laden derart adaptiert, dass sie sich mit dem standardmäßigen Layout dieser Klassen vermischen. Die Defaulttitel `bibliography` und `shorthands` (§ 3.5.7) sind ebenfalls ansprechbar für `\bibintoc` und `\nobibtoc` Wechsel von `memoir`. Die Länge des Registers `\bibitemsep` wird durch `biblatex` auf die gleiche Weise wie `memoir` genutzt (§ 3.8.3). Dieser Abschnitt stellt ebenfalls einige zusätzliche Längeregister vor, welche `\biblistextra` von `memoir` entsprechen. `\setbiblabel` bildet nicht auf eine einzelne Einheit des `biblatex` Pakets ab, da der Stil aller Beschriftungen von dem Bibliographiestil kontrolliert wird. Siehe § 4.2.2 in dem Autorenabschnitt dieser Anleitung für Details. Falls die `memoir`-Klasse entdeckt wurde, stellt `biblatex` ebenso folgende zusätzlichen Tests bereit, welche nützlich für selbstdefinierte Titel sein könnten:

---

`bibliography=nottoc`, `bibliography=toc` und `bibliography=tocnumbered`. Die globalen `toc=bibliography` und `toc=bibliographynumbered` Optionen, und ebenso ihre Aliase, werden ebenso entdeckt. In jedem Fall müssen die Optionen global in das optionale Argument von `\documentclass` gesetzt werden.

```
\ifmemoirbibintoc{<true>}{<false>}
```

Wird zu *<true>* oder *<false>*, abhängig von den `\bibintoc`- und `\nobibintoc`-Schaltern von `memoir`. Dies ist ein LaTeX Frontend für den `\ifnobibintoc`-Test von `memoir`. Man beachte, dass die Logik des Test umgekehrt ist.

### 3.11.3. Seitenzahlen in Zitaten

Falls das *<postnote>* Argument für das Zitierkommando eine Seitenzahl oder ein Seitenbereich ist, wird `biblatex` ihm automatisch ein ‚p.‘ oder ‚pp.‘ voranstellen. Das funktioniert in den typischen Fällen zuverlässig, jedoch ist manchmal manuelles Eingreifen nötig. In diesem Fall ist es wichtig zu verstehen, wie ein Argument ein Detail behandelt. Zuerst checkt `biblatex`, ob die Postnote eine arabische oder römische Zahl ist (case insensitive). Falls dieser Test erfolgreich ist, wird die Postnote als einzelne Seite oder andere Nummer betrachtet, welche mit ‚p.‘ oder einem anderen String präfixt wird, abhängig von dem `pagination`-Feld (siehe § 2.3.10). Falls es fehlschlägt, wird ein zweiter Test durchgeführt, um herauszufinden, ob die Postnote ein Bereich ist oder eine Liste von arabischen oder römischen Zahlen. Falls der Test erfolgreich ist, wird die Postnote mit ‚pp.‘ oder einem String in der Pluralform präfixt. Falls dies ebenfalls fehlschlägt, wird die Postnote gedruckt, wie sie ist. Man beachte, dass beide Tests *<postnote>* erweitern. Alle Kommandos, die in diesem Argument benutzt werden, müssen hierfür robust sein oder mit `\protect` präfixt sein. Hier sind ein paar Beispiele für *<postnote>*-Argumente, die korrekt als einzelne Nummer, einen Bereich von Nummern oder einer Liste von Nummern erkannt wurden:

```
\cite[25]{key} \cite[vii]{key}
\cite[XIV]{key}
\cite[34--38]{key}
\cite[iv--x]{key}
\cite[185/86]{key}
\cite[XI \& XV]{key}
\cite[3, 5, 7]{key}
\cite[vii--x; 5, 7]{key}
```

In einigen anderen Fällen können die Tests falsche Ergebnisse geben und man muss auf die Hilfskommandos `\pno`, `\ppno` und `\nopp` von § 3.6.8 ausweichen. Zum Beispiel, angenommen eine Arbeit ist durch ein besonderes Nummerierungsschema, bestehend aus Nummern und Buchstaben, zitiert. In diesem Schema würde der String ‚27a‘ bedeuten: ‚page 7, part a‘. Da dieser String nicht wie eine Nummer oder ein Bereich für `biblatex` aussieht, muss man das Präfix für eine einzelne Zahl manuell erzwingen:

```
\cite[\pno~27a]{key}
```

Es gibt ebenso ein `\ppno`-Kommando, welches ein Bereichspräfix erzwingt, und ebenso ein `\nopp` Kommando, welches Präfixe unterbindet:

```
\cite[\ppno~27a--28c]{key}
\cite[\nopp 25]{key}
```

Diese Kommandos können überall im `<postnote>`-Argument benutzt werden; auch mehrmals. Zum Beispiel, wenn man mit einem Band und Seitennummer zitiert, möchte man vielleicht das Präfix am Anfang des Postnotes unterbrechen und es in der Mitte des Strings hinzufügen:

```
\cite[VII, \pno~5]{key} \cite[VII, \pno~3,
\ppno~40--45]{key} \cite[see] [\ppno~37--46, in particular \pno~40]{key}
```

Es gibt zwei Hilfskommandos für Suffixe, wie ‚die folgende(n) Seite(n)‘. Statt ein solches Suffix wörtlich einzufügen (was `\ppno` benötigen würde, um ein Präfix zu erzwingen),

```
\cite[\ppno~27~sq.]{key}
\cite[\ppno~55~sq.]{key}
```

benutzt man die Hilfskommandos `\psq` und `\psqq`. Man beachte, dass kein Leerzeichen zwischen Nummer und Kommando ist. Dieses Leerzeichen wird automatisch eingefügt und kann durch Umdefinierung des Makros `\sqspace` modifiziert werden.

```
\cite[27\psq]{key} \cite[55\psqq]{key}
```

Da die Postnote ohne Präfix gedruckt wird, falls es ein Zeichen enthält, welches keine arabische oder römische Zahl ist, kann man das Präfix auch manuell eintippen:

```
\cite[p.~5]{key}
```

Es ist möglich das Präfix über eine pro-Eintrag Basis zu unterdrücken, indem man das `pagination`-Feld von einem Eintrag auf ‚none‘ setzt, siehe § 2.3.10 für Details. Falls man keine Präfixe will oder es bevorzugt, diese manuell einzutippen, kann man ebenso den kompletten Mechanismus in der Dokumentenpräambel oder in der Konfigurationsdatei wie folgt deaktivieren:

```
\DeclareFieldFormat{postnote}{#1}
```

Das `<postnote>`-Argument wird als Feld behandelt und die Formatierung dieses Feldes wird von einer Feldformatierungsrichtlinie kontrolliert, welche frei undefiniert werden kann. Die obere Definitionen druckt die Postnote, so wie sie ist. Siehe §§ ?? and 4.4.2 in der Autorenguide für weitere Details.

### 3.11.4. Namenteile und Namenszwischenräume

Das `biblatex`-Paket bietet den Nutzern und Stilautoren eine sehr gute „gekörnte“ Kontrolle der Namenszwischenräume und des Zeilenumbruchs, besonders wenn Sie Biber als Backend verwenden. Die folgenden diskutierten Befehle sind in §§ 3.8.1 and 4.10.1 dokumentiert. Dieses Kapitel beabsichtigt, einen Überblick über ihre Zusammensetzung zu geben. Ein Hinweis zur Terminologie: Ein Name *part* ist ein grundlegender Teil des Namens, beispielsweise der erste oder letzte Name (Vorname oder Familienname). Jeder Namensteil kann ein einzelner Name oder aus mehreren Teilen zusammengesetzt sein. Beispielsweise, der Namensteil ‚first name‘ (Vorname) kann aus einem ersten und zweiten Vornamen bestehen. Letztere sind Namen-*elements* in diesem Abschnitt. Betrachten wir zuerst einen einfachen Namen: „John Edward Doe“. Dieser Name ist aus folgenden Teilen zusammengesetzt:

```
First John Edward
Prefix —
Last Doe
Suffix —
```

Der Abstand und der Zeilenumbruch wird von vier Makros kontrolliert:

```
a=\bibnamedelima Vom Backend nach dem ersten Element jedes Namensteils
 eingesetzt, wenn dieses Element weniger als drei Zeichen
 lang ist und vor dem letzten Namenselement aller
 Namensteile.
b=\bibnamedelimb Vom Backend zwischen alle Namenteilelemente eingefügt,
 wo \bibnamedelima nicht anwendbar ist.
c=\bibnamedelimc Eingefügt von einer Formatting directive zwischen dem
 Namenspräfix und dem Nachnamen, wenn useprefix=true.
 Wenn useprefix=false, dann ist \bibnamedelimb zu
 nehmen.
d=\bibnamedelimd Eingefügt von einer Formatting directive zwischen
 Namensteilen, wo \bibnamedelimc nicht anwendbar ist.
i=\bibnamedelimi Ersetzt \bibnamedelima/b nach Initialen (großen
 Anfangsbuchstaben).
```

Nun, wie Trennzeichen verwendet werden:

```
JohnEdwardDoe
```

Initialen erhalten in der `bib`-Datei ein spezielles Trennzeichen:

```
J. EdwardDoe
```

Nun wollen wir komplexere Namen betrachten: „Charles-Jean Étienne Gustave Nicolas de La Vallée Poussin“. Diese Namen setzt sich aus folgenden Teilen zusammen:

First Charles-Jean Étienne Gustave Nicolas  
 Prefix de  
 Last La Vallée Poussin  
 Suffix —

Die Trennzeichen:

Charles-Jean<sub>b</sub>Étienne<sub>b</sub>Gustave<sub>a</sub>Nicolas<sub>d</sub>de<sub>c</sub>La<sub>a</sub>Vallée<sub>a</sub>Poussin

Beachten Sie, dass `\bibnamedelima/b/i` durch Backend eingefügt wurde. Das Backend behandelt die Namensteile und kümmert sich um die Trennzeichen zwischen den Elementen, dies baut einen Namensteil um, jeden Namensteil individuell behandelnd. Im Gegensatz dazu werden die Trennzeichen zwischen den teilen eines kompletten Namens (`\bibnamedelimc/d`) entsprechend den *name formatting directives* zu einem späteren Zeitpunkt der Verarbeitungskette hinzugefügt. Abstandserzeugung und Zeichensetzung von Initialen werden auch vom Backend behandelt und kann durch eine Neudefinition der folgenden drei Makros angepasst werden:

|                                    |                                                                                                                                               |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>a=\bibinitperiod</code>      | Eingefügt vom Backend nach Initialen.                                                                                                         |
| <code>b=\bibinitdelim</code>       | Eingefügt vom Backend zwischen mehreren Initialen.                                                                                            |
| <code>c=\bibinithyphendelim</code> | Eingefügt vom Backend zwischen den Initialen eines Bindestrich-Namens, ersetzend <code>\bibinitperiod</code> und <code>\bibinitdelim</code> . |

So werden sie eingesetzt:

J<sub>b</sub>E<sub>a</sub> Doe

K<sub>c</sub>H<sub>a</sub> Mustermann

### 3.11.5. Bibliographiefilter und Zitatbeschriftungen

Die Zitatbeschriftungen, die mit diesem Paket generiert werden, werden in die Liste der Referenzen übertragen, bevor sie von irgendeinem Bibliographiefilter geteilt wird. Sie sind garantiert einzigartig bezüglich des gesamten Dokuments (oder einer `refsection`-Umgebung), egal wie viele Bibliographiefilter genutzt werden. Wenn man ein numerisches Zitateschema benutzt, wird dies meistens zu einer diskontinuierlichen Nummerierung in geteilten Bibliographien führen. Man nutze `defernumbers`-Pakete, um dieses Problem zu umgehen. Falls diese Option aktiviert ist, werden numerische Beschriftungen übertragen, wenn das erste mal ein Eintrag in irgendeiner Bibliographie gedruckt wird.

### 3.11.6. Aktive Zeichen in Bibliographietiteln

Pakete, welche aktive Zeichen benutzen, wie `babel`, `csquotes` oder `underscore`, machen sie normalerweise nicht aktiv bis der Körper des Dokuments beginnt, um Interferenzen mit anderen Paketen zu verhindern. Ein typisches Beispiel eines solchen aktiven Zeichens ist das Ascii zitieren", welches in verschiedenen Sprachmodulen des `babel`-Pakets benutzt wird. Falls Kürzel wie "<" und "a" in einem Argument für `\defbibheading` benutzt werden und die Titel in der Dokumentenpräambel definiert sind, ist die nichtaktive Form des Zeichens in der Titeldefinition gesichert. Wenn der Titel ein Schriftsatz ist, funktionieren sie nicht als Kommandos, sondern werden einfach wörtlich gedruckt. Die direkteste Lösung besteht darin `\defbibheading` hinter `\begin{document}` zu verschieben. Alternativ kann man die `\shorthandon` und `\shorthandoff` Kommandos von `babel` nutzen, um die Kürzel temporär in der Präambel zu aktivieren. Das oben bezieht sich ebenfalls auf Bibliographieanmerkungen und das `\defbibnote` Kommando.

### 3.11.7. Gruppierung in Referenzabschnitten und -segmenten

Alle LaTeX-Umgebungen, umgeben von `\begin` und `\end`, formen eine Gruppe. Dies könnte unerwünschte Nebeneffekte haben, falls die Umgebung etwas beinhaltet, dass nicht dazu gedacht ist, in einer Gruppe benutzt zu werden. Dieses Problem ist nicht spezifisch für `refsection` und `refsegment` Umgebungen, allerdings gilt es offensichtlich auch für sie. Da diese Umgebungen normalerweise einen größeren Anteil des Dokuments umschließen als eine typische `itemize`- oder ähnliche Umgebung, rufen sie mit höherer Wahrscheinlichkeit Probleme bezüglich Gruppierungen hervor. Falls man Fehlfunktionen beobachtet, nachdem man `refsection`-Umgebungen zu einem Dokument hinzufügte (zum Beispiel, falls irgendetwas in der Umgebung ‚gefangen‘ zu sein scheint), probiere man stattdessen folgende Syntax:

```
\chapter{...}
\refsection ...
\endrefsection
```

Dies wird keine Gruppe erzeugen, aber funktioniert sonst normal. Soweit `biblatex` betroffen ist, ist es egal, welche Syntax man benutzt. Die alternative Syntax wird ebenso von der `refsegment`-Umgebung unterstützt. Man beachte, dass die Kommandos `\newrefsection` und `\newrefsegment` keine Gruppe erzeugen. Siehe §§ 3.5.4 and 3.5.5 für Details.

## 4. Autorenanleitung

Dieser Teil der Anleitung dokumentiert das Autoreninterface des `biblatex`-Paketes. Die Autorenanleitung deckt alles ab, was man wissen muss, um einen neuen Zitat- und Bibliographiestil zu schreiben oder für Lokalisierungsmodule. Man sollte die User Guide zu erst lesen, bevor man mit diesem Teil der Anleitung beginnt.

### 4.1. Überblick

Bevor wir zu den Kommandos und Hilfen kommen, welche von `biblatex` bereitgestellt werden, werfen wir einen Blick auf einige fundamentale Konzepte. Das `biblatex`-Paket wirkt mit BibTeX auf eine Weise zusammen, die etwas anders ist als die traditioneller Weise. Am Wichtigsten ist, dass die `bb1`-Datei anders genutzt wird und es nur eine `bst`-Datei gibt, welche eher ein strukturiertes Dateninterface implementiert, als druckbare Daten exportiert. Mit den Standardbibliographiehilfen von LaTeX beinhaltet ein Dokument eine beliebige Anzahl von Zitierkommandos im Dokumentenkörper plus `\bibliographystyle` und `\bibliography` normalerweise zum Ende des Dokuments. Der Ort des Ersteren ist beliebig, das Letztere markiert den Ort, wo die Liste der Referenzen gedruckt werden soll:

```
\documentclass{...}
\begin{document}
\cite{...} ...
\bibliographystyle{...}
\bibliography{...}
\end{document}
```

Die Verarbeitung dieser Datei benötigt eine bestimmte Prozedur als Nachfolger. Diese Prozedur ist Folgende:

1. Durchlaufen von `latex`: Im ersten Lauf werden `\bibstyle` und `\bibdata` Kommandos in die `aux`-Datei geschrieben, zusammen mit `\citation` Kommandos für alle Zitate. An diesem Punkt sind die Referenzen undefiniert, weil LaTeX auf BibTeX wartet, welches die benötigten Daten bereitstellen soll. Es gibt auch noch keine Bibliographie.
2. Durchlaufen von `bibtex`: BibTeX schreibt eine `thebibliography`-Umgebung in die `bb1`-Datei und stellt so alle Einträge der `bib`-Datei bereit, welche von den `\citation` Kommandos in der `aux`-Datei angefragt wurden.
3. Durchlaufen von `latex`: Beginnend mit dem zweiten Lauf, schreiben die `\bibitem` Kommandos in der `thebibliography`-Umgebung ein `\bibcite` Kommando für jeden Bibliographieeintrag in die `aux`-Datei. Diese `\bibcite` Kommandos

definieren die Zitatbeschriftungen, welche durch `\cite` benutzt werden. Die Referenzen sind immer noch undefiniert, weil die Beschriftungen nicht vor dem Ende des Laufes verfügbar sind.

4. Durchlaufen von `latex`: Beginnend mit dem dritten Lauf, werden Zitatbeschriftungen so definiert, wie sie in der `aux`-Datei am Ende der Präambel eingelesen werden. Alle Zitate können nun gedruckt werden.

Man beachte, dass alle Bibliographiedaten in die `bbl`-Datei in ihrem finalen Format geschrieben sind. Die `bbl`-Datei wird, wie jede druckbare Sektion eines Dokuments, eingelesen und verarbeitet. Zum Beispiel betrachte man folgenden Eintrag in einer `bib`-Datei:

```
@Book{companion,
author = {Michel Goossens and Frank Mittelbach and Alexander Samarin},
title = {The LaTeX Companion},
publisher = {Addison-Wesley},
address = {Reading, Mass.},
year = {1994},
}
```

Mit dem `plain.bst`-Stil exportiert BibTeX diesen Eintrag, wie folgt, in die `bbl` Datei:

```
\bibitem{companion} Michel Goossens, Frank Mittelbach, and Alexander Samarin.
\newblock {\em The LaTeX Companion}.
\newblock Addison-Wesley, Reading, Mass., 1994.
```

Standardmäßig generiert LaTeX numerische Zitatbeschriftungen, so dass `\bibitem` Zeilen, wie folgende, in die `aux`-Datei schreibt:

```
\bibcite{companion}{1}
```

Die Umsetzung eines anderen Zitatestils impliziert, dass mehr Daten durch die `aux`-Datei transferiert werden müssen. Mit dem `natbib`-Paket, zum Beispiel, beinhaltet die `aux`-Datei Zeilen wie diese:

```
\bibcite{companion}{{1}{1994}{{Goossens et-al.}}
{{Goossens, Mittelbach, and Samarin}}}
```

Das `biblatex`-Paket unterstützt Zitate in jedem beliebigen Format, so dass Zitierkommandos Zugang zu allen Bibliographiedaten benötigt. Was dies im Rahmen der Prozedur, wie oben umrissen, bedeuten würde, wird offensichtlich, wenn man auf die Ausgabe des `jurabib`-Pakets schaut, welches ebenso alle Bibliographiedaten in Zitaten zugänglich macht:

```

\bibcite{companion}{{Goossens \jbbfsasep Mittelbach\jbbstasep Samarin}%
}{0}{book}{1994}{}{}{}{Reading, Mass.\bpubaddr{}Addison-Wesley%
\bibbdsep{} 1994}{{The LaTeX Companion}{}{}{2}{}{}{}{}{}{}
{\bibnf {Goossens}{Michel}{M.}{}{}\Bibbfsasep\bibnf{Mittelbach}{Frank}{F.}%
{}{}\Bibbstasep\bibnf{Samarin}{Alexander}{A.}{}{}
{\bibtfnt{The LaTeX Companion}.\
\apyformat{Reading, Mass.\bpubaddr{} Addison-Wesley\bibbdsep{} 1994}}}

```

In diesem Fall sind alle Inhalte der kompletten `thebibliography`-Umgebung effektiv durch die `aux`-Datei transferiert. Die Daten werden von der `bbl`-Datei eingelesen und in die `aux`-Datei geschrieben und anschließend wieder von der `aux`-Datei eingelesen und im Speicher behalten. Die Bibliographie selber wird noch generiert, wenn die `bbl`-Datei eingelesen wird. Das `biblatex`-Paket wird ebenso gezwungen werden, alle Daten durch die `aux`-Datei laufen zu lassen. Dies impliziert hängende Verarbeitungen und ist ebenso redundant, weil die Daten sowieso im Speicher behalten werden müssen.

Die traditionelle Prozedur basiert auf der Annahme, dass die gesamten Bibliographiedaten eines Eintrags nur in der Bibliographie benötigt werden und dass alle Zitate kurze Beschriftungen nutzen. Dies macht es sehr effektiv bezüglich Speicherbedarf, aber es impliziert ebenso, dass es nicht gut skaliert. Darum nimmt `biblatex` einen andere Ansatz. Zunächst ist die Dokumentenstruktur etwas anders. Statt `\bibliography` im Dokumentenkörper zu nutzen, sind Datenbankdateien in der Präambel mit `\addbibresource` spezifiziert, `\bibliographystyle` wird komplett weggelassen (alle Funktionen werden durch die Paketoptionen kontrolliert) und die Bibliographie wird durch `\printbibliography` gedruckt:

```

\documentclass{...}
\usepackage[...]{biblatex}
\addbibresource{...}
\begin{document} \cite{...}
...
\printbibliography
\end{document}

```

Um die Prozedur zu modernisieren, verwendet `biblatex` die `bbl`-Datei wie eine `aux`-Datei und erklärt somit `\bibcite` als veraltet. Man bekommt somit die folgende Prozedur:

1. Durchlaufen von `latex`: Der erste Schritt ist ähnlich zu der traditionellen Prozedur, wie oben beschrieben: `\bibstyle` und `\bibdata` Kommandos werden in die `aux`-Datei geschrieben, zusammen mit den `\citation` Kommandos für alle Zitate. Wir warten dann auf BibTeX, welches die benötigten Daten bereitstellt. Da `biblatex` eine spezielle `bst`-Datei verwendet, welche sein Dateninterface

auf das BibTeX Backend implementiert, ist das `\bibstyle` Kommando immer `\bibstyle{biblatex}`.

2. Durchlaufen von `bibtex`: BibTeX stellt die Einträge von der `bib`-Datei bereit, welche von den `\citation` Kommandos in der `aux`-Datei angefragt wurden. Es schreibt keine druckbare Bibliographie in die `bb1`-Datei, eher eine strukturierte Darstellung der Bibliographiedaten. Genau wie eine `aux`-Datei druckt die `bb1`-Datei nichts, wenn es einliest. Es speichert lediglich Daten im Speicher.
3. Durchlaufen von `latex`: Beginnend mit dem zweiten Lauf, wird die `bb1`-Datei direkt am Anfang des Dokumentenkörpers verarbeitet, genauso wie eine `aux`-Datei. Von diesem Punkt an sind die Bibliographiedaten im Speicher verfügbar, so dass alle Zitate direkt gedruckt werden können.<sup>26</sup> Die Zitierkommandos haben Zugang zu den kompletten Bibliographiedaten, nicht nur zu einer vordefinierten Beschriftung. Die Bibliographie wird aus dem Speicher durch Nutzung der selben Daten generiert und kann wie gefordert gefiltert oder geteilt werden.

Betrachten wir den Beispieleintrag von oben nochmal:

```
@Book{companion,
author = {Michel Goossens and Frank Mittelbach and Alexander Samarin},
title = {The LaTeX Companion},
publisher = {Addison-Wesley},
address = {Reading, Mass.},
year = {1994},
}
```

Mit `biblatex` wird dieser Eintrag im Wesentlichen in das folgende Format exportiert:

```
\entry{companion}{book}{}
\name{author}{3}{%
{{Goossens}{G.}{Michel}{M.}{}{}{}{}}%
{{Mittelbach}{M.}{Frank}{F.}{}{}{}{}}%
{{Samarin}{S.}{Alexander}{A.}{}{}{}{}}%
}
\list{publisher}{1}{%
{Addison-Wesley}%
}
\list{location}{1}{%
```

<sup>26</sup>Falls die `defernumbers` Paketooption aktiviert ist, benutzt `biblatex` einen Algorithmus ähnlich zu der traditionellen Prozedur, um numerische Beschriftungen zu generieren. In diesem Fall werden die Zahlen übertragen, sobald die Bibliographie gedruckt ist und dann durchlaufen sie die `aux`-Datei. Es wird ein zusätzlicher LaTeX-Lauf für sie benötigt, um in die Zitate aufgenommen zu werden.

```
{Reading, Mass.}%
}
\field{title}{The LaTeX Companion}
\field{year}{1994}
\endentry
```

Wie in diesem Beispiel gesehen, werden die Daten in einem strukturierten Format präsentiert, welches ähnlich zu der Struktur einer bib-Datei ist. An diesem Punkt wurde keine Entscheidung bezüglich des finalen Formats des Bibliographieeintrags gemacht. Die Formatierung der Bibliographie und aller Zitate wird durch LaTeX-Makros kontrolliert, welche in der Bibliographie- und Zitate-Stildateien definiert sind.

## 4.2. Bibliographiestile

Ein Bibliographiestil ist ein Satz (Set) von Makros, welche die Einträge in die Bibliographie drucken. Solche Stile sind in Dateien mit dem Suffix bbx definiert. Das biblatex-Paket lädt die gewählten Bibliographie-Stil-Dateien am Ende des Pakets. Man beachte, dass eine kleine Auswahl von häufig genutzten Makros, welche von einigen Standard-Bibliographie-Stilen geteilt werden, in biblatex.def enthalten sind. Diese Datei wird ebenso am Ende des Pakets, vor dem ausgewählten Bibliographiestil, geladen.

### 4.2.1. Bibliographie-Stildateien

Bevor wir zu den einzelnen Komponenten von einem Bibliographiestil übergehen, betrachten wir dieses Beispiel einer allgemeinen Struktur von einer typischen bbx-Datei:

```
\ProvidesFile{example.bbx}[2006/03/15 v1.0 biblatex bibliography style]

\defbibenvironment{bibliography} {...} {...} {...}
\defbibenvironment{shorthands} {...} {...} {...}
\InitializeBibliographyStyle{...}
\DeclareBibliographyDriver{article}{...}
\DeclareBibliographyDriver{book}{...}
\DeclareBibliographyDriver{inbook}{...}
...
\DeclareBibliographyDriver{shorthands}{...}
\endinput
```

Die Hauptstruktur einer Bibliographie-Stil-Datei besteht aus folgenden Kommandos:

`\RequireBibliographyStyle{<style>}`

Dieses Kommando ist optional und ist für spezialisierte Bibliographiestile gedacht, welche auf einem allgemeineren Stil aufbauen. Es lädt den Bibliographiestil `style`. `bbx`.

`\InitializeBibliographyStyle{<code>}`

Definiert beliebigen `<code>`, so dass er am Anfang der Bibliographie eingefügt wird, jedoch innerhalb der Gruppe, welche von der Bibliographie geformt wird. Dieses Kommando ist optional. Es kann nützlich sein für Definitionen, welche von einigen Bibliographietreibern geteilt werden, aber nicht außerhalb der Bibliographie genutzt werden. Man merke sich, dass es mehrere Bibliographien in einem Dokument geben kann. Falls die Bibliographietreiber eine globale Anweisung erzeugen, sollten sie am Anfang der nächsten Bibliographie gelöscht werden.

`\DeclareBibliographyDriver{<entrytype>}{<code>}`

Definiert einen Bibliographietreiber. Ein ‚Treiber‘ ist ein Makro, welches spezielle Eintragstypen bearbeitet. Der `<entrytype>` entspricht dem Eintragstypen, welcher in `bib`-Dateien benutzt wurde, spezifiziert in Kleinbuchstaben (siehe § 2.1). Der Typ `shorthands` hat eine besondere Bedeutung. Der `shorthands`-Treiber bearbeitet alle Einträge in einer Liste von Kürzeln. Das `<entrytype>`-Argument kann ebenso ein Sternchen sein. In diesem Fall dient der Treiber als ein Rückgriff, welcher benutzt wird, falls kein spezifischer Treiber für den Eintragstypen definiert wurde. Der `<code>` ist ein beliebiger Kode, welcher alle Bibliographieeinträge des entsprechenden `<entrytype>` setzt. Dieses Kommando obligatorisch. Jeder Bibliographie-Style sollte einen Treiber für jeden Eintragstypen bereitstellen.

`\DeclareBibliographyAlias{<alias>}{<entrytype>}`

Falls ein Bibliographietreiber mehr als einen Eintragstypen abdeckt, kann dieses Kommando benutzt werden, um einen Alias zu definieren. Dieses Kommando ist optional. Das `<entrytype>`-Argument kann ebenso ein Sternchen sein. In diesem Fall dient der Alias als ein Rückgriff, welcher benutzt wird, falls kein spezifischer Treiber für diesen Eintragstypen definiert wurde.

`\DeclareBibliographyOption{<key>}[<value>]{<code>}`

Dieses Kommando definiert zusätzliche Präambeloptionen im `<key>=<value>`-Format. Der `<key>` ist ein Optionsschlüssel. Der `<code>` ist beliebiger TeX-Kode, welcher ausgeführt werden soll, wenn die Option benutzt wird. Der Wert, welcher der Option übergeben wurde, wird als #1 weiter an den `<code>` gegeben. Der optionale `<value>` ist ein Defaultwert, welcher benutzt wird, falls der blanke Schlüssel ohne einen Wert übergeben wurde. Dies ist nützlich für boolesche Schalter. Zum Beispiel mit einer Definition wie der Folgenden:

```
\DeclareBibliographyOption{somekey}[true]{...}
```

Einen ‚somekey‘ ohne Wert zu geben, ist äquivalent zu ‚somekey=true‘.

```
\DeclareEntryOption{<key>}[<value>]{<code>}
```

Ähnlich zu `\DeclareBibliographyOption`, aber definiert Optionen, welche auf einer pro-Eintrag Basis in dem options Feld von § 2.2.3 setzbar sind. Der `<code>` wird immer dann ausgeführt, wenn `biblatex` die Daten eines Eintrags zur Nutzung durch ein Zitierkommando oder ein Bibliographietreiber vorbereitet.

#### 4.2.2. Bibliographieumgebungen

Abgesehen vom Definieren von Bibliographietreibern, ist der Bibliographie-Stil auch für die Umgebungen verantwortlich, die das Layout der Bibliographie und die Liste der Kürzel kontrollieren. Diese Umgebungen werden mit `\defbibenvironment` definiert. Standardmäßig nutzt `\printbibliography` die Umgebung `bibliography`. Hierbei ist eine Definition passend für einen Bibliographiestil, welche keine Beschriftungen in die Bibliographie druckt:

```
\defbibenvironment{bibliography} {\list {}
{\setlength{\leftmargin}{\bibhang}%
\setlength{\itemindent}{-\leftmargin}%
\setlength{\itemsep}{\bibitemsep}%
\setlength{\parsep}{\bibparsep}}}
{\endlist} {\item}
```

Diese Definition nutzt eine `list`-Umgebung mit hängender Markierung unter Verwendung des `\bibhang`-Längenregisters, welcher durch `biblatex` bereitgestellt wird. Es erlaubt zu einem gewissen Grad die Konfigurierbarkeit durch Nutzung von `\bibitemsep` und `\bibparsep`, zwei Längenregister, welche durch `biblatex` für diesen Zweck bereitgestellt werden (siehe § 4.10.3). Der `authoryear` und `authortitle` Bibliographiestil nutzt eine Definition ähnlich zu diesem Beispiel.

```
\defbibenvironment{bibliography}
{\list {\printfield[labelnumberwidth]{labelnumber}}
{\setlength{\labelwidth}{\labelnumberwidth}%
\setlength{\leftmargin}{\labelwidth}%
\setlength{\labelsep}{\biblabelsep}%
\addtolength{\leftmargin}{\labelsep}%
\setlength{\itemsep}{\bibitemsep}%
\setlength{\parsep}{\bibparsep}}%
\renewcommand*{\makelabel}[1]{\hss##1}}
{\endlist}
{\item}
```

Manche Bibliographiestile drucken Beschriftungen in die Bibliographie. Zum Beispiel, ein Bibliographiestil, der für ein numerisches Zitierschema ausgelegt ist, wird die Nummer von jedem Eintrag so drucken, dass die Bibliographie wie eine nummerierte Liste aussieht. In dem ersten Beispiel war das erste Argument für `\list` leer. In diesem Beispiel brauchen wir es, um eine Nummer einzufügen, welche durch `biblatex` im `labelnumber`-Feld bereitgestellt wird. Wir nutzen ebenso einige Längenregister und andere Einheiten, welche durch `biblatex` bereitgestellt werden, siehe §§ 4.10.4 und 4.10.5 für Details. Der `numeric`-Bibliographiestil nutzt die oben gegebene Definition. Der `alphabetic`-Stil ist ähnlich, außer, dass `labelnumber` durch `labelalpha` und `labelnumberwidth` durch `labelalphawidth` ersetzt werden.

Diese Liste von Kürzeln wird in einer ähnlichen Weise bearbeitet.

`\printshorthands` nutzt standardmäßig die `shorthands`-Umgebung. Ein typisches Beispiel ist unten gegeben. Siehe §§ 4.10.4 und 4.10.5 für Details zu Längenregistern und Einheiten, welche in diesem Beispiel benutzt werden.

```
\defbibenvironment{shorthands}
{\list
{\printfield[shorthandwidth]{shorthand}}
{\setlength{\labelwidth}{\shorthandwidth}%
\setlength{\leftmargin}{\labelwidth}%
\setlength{\labelsep}{\biblabelsep}%
\addtolength{\leftmargin}{\labelsep}%
\setlength{\itemsep}{\bibitemsep}%
\setlength{\parsep}{\bibparsep}%
\renewcommand*{\makelabel}[1]{##1\hss}}
{\endlist}
{\item}
```

### 4.2.3. Bibliographietreiber

Bevor wir zu Kommandos übergehen, die das Dateninterface für das `biblatex`-Paket formen, kann es lehrreich sein, einen Blick auf die Struktur eines Bibliographietreibers zu werfen. Man beachte, dass das Beispiel unten sehr vereinfacht ist, aber immer noch funktionsfähig. Für die Lesbarkeit, lassen wir einige Felder weg, welche Teil eines `@book`-Eintrags sein können und vereinfachen ebenso die Handhabung von denen, die betrachtet werden. Der Hauptpunkt ist, eine Vorstellung zu geben, wie ein Treiber strukturiert ist. Für Informationen über die Zuordnung von BibTeX Feldern zu `biblatex`-Datentypen, siehe § 2.2.

```
\DeclareBibliographyDriver{book}{%
\printnames{author}%
\newunit\newblock \printfield{title}%
\newunit\newblock \printlist{publisher}%
```

```

\newunit \printlist{location}%
\newunit \printfield{year}%
\finentry}

```

An diesem Punkt fehlt immer noch ein Stück: Die Formatierungsrichtlinien, welche von `\printnames`, `\printlist` und `\printfield` genutzt werden. Um einen Einblick zu geben, wie eine Formatierungsrichtlinie aussieht, sind hier einige Fiktionale, welche von unserem Beispieldreiber genutzt wurden. Feldformate sind einfach, da der Wert des Feldes der Formatierungsrichtlinie als Argument übergeben wird, welches wie gewünscht formatiert werden kann. Die folgende Richtlinie wird einfach sein Argument in ein `\emph` Kommando packen:

```

\DeclareFieldFormat{title}{\emph{#1}}

```

Listenformate sind etwas komplexer. Nachdem die Liste in einzelne Einheiten geteilt wurde, wird `biblatex` die Formatierungsrichtlinie einmal auf jede Einheit in der Liste ausführen. Die Einheit wird zu der Richtlinie als ein Argument übergeben. Der Separator, welcher zwischen die einzelnen Einheiten in der Liste eingefügt werden soll, wird ebenso durch die entsprechende Richtlinie bearbeitet, wodurch man prüfen muss, ob man in der Mitte der Liste oder am Ende ist, wenn man ihn einfügt.

```

\DeclareListFormat{location}{%
#1%
\ifthenelse{\value{listcount}<\value{liststop}} {\addcomma\space} {}}

```

Formatierungsrichtlinien für Namen sind ähnlich zu denen für eigentliche Listen, aber die einzelnen Einheiten in der Liste sind Namen, welche automatisch in ihre Einzelteile zerlegt werden. Die Listenformatierungsrichtlinie wird einmal für jeden Namen in der Liste ausgeführt und die Komponenten des Namens werden an die Formatierungsrichtlinie als getrennte Argumente übergeben. Zum Beispiel, `#1` ist der letzte Name und `#3` ist der erste Name. Hier ist ein vereinfachtes Beispiel:

```

\DeclareNameFormat{author}{%
\ifthenelse{\value{listcount}=1} {#1%
\ifblank{#3}{}{\addcomma\space #3}}
{\ifblank{#3}{}{#3\space}% #1}%
\ifthenelse{\value{listcount}<\value{liststop}}
{\addcomma\space} {}}

```

Die Richtlinie oben dreht die Namen des ersten Autor („Last, First“) herum und druckt die übrigen Namen in der regulären Reihenfolge („First Last“). Man beachte, dass die einzige Komponente, welche garantiert verfügbar ist, der Nachname ist, wodurch man prüfen muss, welche Teile des Namens tatsächlich präsent sind. Falls eine bestimmte Komponente des Namens nicht verfügbar ist, ist das entsprechende

Argument leer. Wie bei Richtlinien für wörtliche Listen wird der Separator, welcher zwischen den einzelnen Einheiten in der Liste eingefügt werden soll, ebenso durch die Formatierungsrichtlinie bearbeitet, wodurch man prüfen muss, ob man in der Mitte oder am Ende der Liste ist, wenn man ihn einfügt. Dies ist das, was der zweite `\ifthenelse`-Test tut.

#### 4.2.4. Spezialfelder

Die folgenden Listen und Felder werden von `biblatex` benutzt, um Daten an die Bibliographietreiber und Zitierkommandos zu übergeben. Sie werden nicht in `bib`-Dateien benutzt, aber automatisch durch das Paket definiert. Von der Perspektive eines Bibliographie- oder Zitiertiles sind sie nicht verschieden von den Feldern in der `bib`-Datei.

##### 4.2.4.1. Allgemeine Felder

`entrykey` field (string)

Der Eintragsschlüssel von einem Element in der `bib`-Datei. Dies ist der String, der von `biblatex` und `BibTeX` benutzt wird, um einen Eintrag in der `bib`-Datei zu identifizieren.

`childentrykey` field (string)

Wenn man einen Untereintrag von einem Eintragsatz zitiert, stellt `biblatex` Daten von dem Eltern-`@set`-Eintrag für die Zitierkommandos bereit. Dies impliziert, dass das `entrykey`-Feld den Eintragsschlüssel von den Eltern beinhaltet. Der Eintragsschlüssel von dem zitierten Kindeintrag wird in dem `childentrykey` Feld bereitgestellt. Dieses Feld ist nur verfügbar, wenn ein Untereintrag eines Eintragsatzes zitiert wird.

`entrytype` field (string)

Der Eintragstyp (`@book`, `@inbook`, etc.), gegeben in Kleinbuchstaben.

`childentrytype` field (string)

Wenn ein Untereintrag von einem Eintragsatz zitiert wird, stellt `biblatex` die Daten des Eltern-`@set`-Eintrags für Zitierkommandos bereit. Dies impliziert, dass das `entrytype` Feld den Eintragstyp von den Eltern beinhaltet. Der Eintragstyp von dem zitierten Kindeintrag wird in dem `childentrytype`-Feld bereitgestellt. Dieses Feld ist nur verfügbar, wenn ein Untereintrag eines Eintragsatzes zitiert wird.

**entrysetcount** field (integer)

Dieses Feld beinhaltet einen Integer, welcher die Position von einem Satzmitglied in einem Eintragsatz anzeigt (startend mit 1). Dieses Feld ist nur in den Untereinträgen eines Eintragsatzes verfügbar.

**hash** field (string)

Biber only

Dieses Feld ist besonders, da es nur für lokal vorhandene Richtlinien zur Namensformatierung ist. Es trägt einen Hash-String, der Namen aus der Namensliste eindeutig identifiziert. Diese Information steht für alle Namen in allen Namenslisten zur Verfügung. Beachten Sie auch die Felder `namehash` und `fullhash`.

**namehash** field (string)

Ein einzigartiger Hashstring, abgeleitet von der `labelname`-Liste. Dieser ist für wiederkehrende Checks. Zum Beispiel, ein Zitatestil, welcher einen wiederkehrenden Autoren oder Verfasser mit einem String wie ‚idem‘ ersetzt, könnte das `namehash`-Feld mit `\savefield` sichern und es später im Vergleich mit `\iffieldequals` nutzen (siehe §§ 4.6.1 and 4.6.2). Der `namehash` ist abgeleitet von der beschränkten `labelname`-Liste, z. B. ist es ansprechbar für die globalen `maxnames` und `minnames` Optionen. Es ist automatisch so gereinigt, dass der Hash sicher in Formationen von Kontrollsequenznamen genutzt werden kann. Es ist nicht zum Drucken gedacht. Siehe auch `fullhash`.

**fullhash** field (string)

Ein Hash-String, der eine eindeutige Identifizierung der `labelname`-Liste vornimmt. Diese Felder unterscheiden sich von `namehash` in zwei Details: 1. Die `shortauthor`- und `shorteditor`-Listen werden ignoriert, wenn der Hash erzeugt wird. 2. Der Hash bezieht sich immer auf die ganze Liste, ignorierend `maxnames` und `minnames`. Beachten Sie auch `hash` und `namehash`.

**pageref** list (literal)

Falls die `backref` Paketooption aktiviert ist, beinhaltet diese Liste die Seitennummern von Seiten, auf denen betreffende Bibliographieeinträge zitiert werden. Falls es `refsection` Umgebungen im Dokument gibt, sind die Rückreferenzen lokal für die Referenzabschnitte.

**sortinit** field (literal)

Dieses Feld beinhaltet den ersten Buchstaben von dem String, welcher während der Sortierung genutzt wird. Dies ist nützlich, wenn man die Bibliographie alphabetisch unterteilt.

#### 4.2.4.2. Felder für die Nutzung in Zitatbeschriftungen

`labelalpha` field (literal)

Eine Beschriftung ähnlich zu den Beschriftungen, welche von dem `alpha.bst`-Stil von dem traditionellen BibTeX generiert werden. Diese Beschriftungen bestehen aus dem Anfangsbuchstaben, bezogen aus der `labelname`-Liste, plus die letzten zwei Stellen des Publikationsjahres. Das `label`-Feld kann genutzt werden, um seinen nicht-nummerischen Teil zu überschreiben. Falls das `label`-Feld definiert ist, wird `biblatex` seinen Wert nutzen und die letzten zwei Stellen vom Publikationsjahr anhängen, wenn es `labelalpha` generiert. Das `shorthand`-Feld kann genutzt werden, um die ganze Beschriftung zu überschreiben. Falls definiert, ist `labelalpha` eher ein `shorthand` statt eine automatisch generierte Beschriftung. Eine komplette ‚alphabetische‘ Beschriftung besteht aus den Feldern `labelalpha` plus `extraalpha`. Man beachte, dass die `labelalpha`- und `extraalpha`-Felder mit der Paketoption `labelalpha` angefragt werden müssen (§ 3.1.2.3). Siehe auch `extraalpha` und `\labelalphaothers` in § 3.8.1.

`extraalpha` field (integer)

Das ‚alphabetische‘ Zitierschema benötigt normalerweise einen Buchstaben zum Anhängen an die Beschriftung, falls die Bibliographie zwei oder mehr Arbeiten des selben Autors beinhaltet, welche im selben Jahr publiziert wurden. In diesem Fall beinhaltet das `extraalpha`-Feld ein Integer, welcher in einen Buchstaben mit `\mknumalph` konvertiert werden kann oder auf eine andere Weise formatiert werden kann. Dieses Feld ist ähnlich zu der Rolle von `extrayear` in dem Autor-Jahr-Schema. Eine komplette ‚alphabetische‘ Beschriftung besteht aus den Feldern `labelalpha` plus `extraalpha`. Man beachte, dass die `labelalpha`- und `extraalpha`-Felder durch die Paketoption `labelalpha` angefordert werden müssen, siehe § 3.1.2.3 für Details. Siehe auch `labelalpha` und `\labelalphaothers` in § 3.8.1.

`labelname` list (name)

Der Name, der in Zitaten gedruckt werden soll. Diese Liste ist entweder eine Kopie von `shortauthor`, `author`, `shorteditor`, `editor` oder von der `translator`-Liste, welche normalerweise in dieser Reihenfolge gecheckt werden. Falls kein Autor oder Verfasser verfügbar ist, ist diese Liste undefiniert. Man beachte, dass diese Liste auch für die `useauthor`-, `useeditor`- und `usetranslator`-Optionen ansprechbar ist, siehe § 3.1.3. Zitatestile sollten diese Liste nutzen, wenn sie die Namen in die Zitate drucken. Diese Liste wird nur für die Bequemlichkeit bereitgestellt und beinhaltet keine zusätzliche Information.

Mit Biber kann dieses Feld angepasst werden. Siehe § 4.5.2 für Details.

Biber only

**labelnumber** field (literal)

Die Nummer der Bibliographieeinträge, wie sie für das numerische Zitierschema benötigt wird. Falls das `shorthand`-Feld definiert ist, bestimmt `biblatex` keine Nummer für den entsprechenden Eintrag. In diesem Fall ist `labelnumber` eher ein Kürzel statt eine Nummer. Numerische Stile müssen den Wert von diesem Feld nutzen statt einen Zähler. Man beachte, dass dieses Feld mit der Paketooption `labelnumber` angefordert werden muss, siehe § 3.1.2.3 für Details. Siehe auch die Paketooption `defernumbers` in § 3.1.2.1.

**prefixnumber** field (literal)

Falls die `prefixnumbers` Option von `\printbibliography` gesetzt wurde, um allen Einträgen in einer Unterbibliographie ein Präfix in Form eines festen Strings voranzustellen, ist dieser String in dem `prefixnumber`-Feld von allen betroffenen Einträgen verfügbar. Falls kein Präfix gesetzt wurde, ist das `prefixnumber`-Feld des entsprechenden Eintrags undefiniert. Siehe die `prefixnumbers` Option von `\printbibliography` in § 3.5.2 für Details. Falls das `shorthand`-Feld definiert ist, bestimmt `biblatex` kein Präfix für das `prefixnumber`-Feld des entsprechenden Eintrags. In diesem Fall ist das `prefixnumber`-Feld undefiniert.

**labeltitle** field (literal)

Der Titel, der in den Zitaten gedruckt werden soll. Falls ein Kurztitel verfügbar ist, ist `labeltitle` eine Kopie vom `shorttitle`-Feld, anderenfalls greift `biblatex` auf das `title`-Feld zurück. Zitierstile, welche Kurztitel drucken, können einfach dieses Feld nutzen, statt jedesmal, wenn sie einen Titel drucken, zu checken, ob ein `shorttitle`-Feld verfügbar ist. Dieses Feld wird nur für die Bequemlichkeit bereitgestellt und enthält keine zusätzliche Information.

**labelyear** field (literal)

Das Publikationsjahr, wie es in dem `date`- oder `year`-Feld spezifiziert ist, für die Nutzung in Autor-Jahr-Beschriftungen. Eine komplette Autor-Jahr-Beschriftung besteht aus den Feldern `labelyear` plus `extrayear`. Man beachte, dass die `labelyear`- und `extrayear`-Felder von der Paketooption `labelyear` angefordert werden müssen, siehe § 3.1.2.3 für Details. Siehe auch `extrayear`.

Mit Biber kann dieses Feld angepasst werden. Siehe § 4.5.2 für Details.

Biber only

**extrayear** field (integer)

Das Autor-Jahr-Zitierschema benötigt normalerweise einen Buchstaben zum anhängen an das Jahr, falls die Bibliographie zwei oder mehr Arbeiten von dem selben

Autor enthält, welche im gleichen Jahr veröffentlicht wurden. In diesem Fall beinhaltet das `extrayear`-Feld einen Integer, welcher durch `\mknumalph` konvertiert werden kann oder durch einen anderen Weg. Dieses Feld ist undefiniert, falls es nur eine Arbeit von dem Autoren in der Bibliographie gibt oder falls alle Arbeiten von einem Autor unterschiedliche Publikationsjahre haben. Eine komplette Autor-Jahr-Beschriftung besteht aus den Feldern `labelyear` plus `extayear`. Man beachte, dass die `labelyear`- und `extrayear`-Felder von der Paketoption `labelyear` angefordert werden müssen, siehe § 3.1.2.3 für Details. Siehe auch `labelyear`.

**4.2.4.3. Datumsfeld** Siehe Tabelle 7 für einen Überblick, wie die Datumsfelder in `bib`-Dateien mit den Datumsfeldern vom Stilinterface verwandt sind. Wenn man ein Feld wie `origdate` in einem Stil testet, nutze man einen Kode wie:

```
\iffieldundef{origyear}{...}{...}
```

Dieser wird zeigen, ob das entsprechende Datum überhaupt definiert ist. Dieser Test

```
\iffieldundef{origendyear}{...}{...}
```

wird zeigen, ob das entsprechende Datum und ein (voll spezifizierter) Bereich definiert ist. Dieser Test:

```
\iffieldequalstr{origendyear}{...}{...}
```

wird zeigen, ob das entsprechende Datum definiert ist und ob es ein offener Bereich ist. Offene Bereiche werden durch leere `endyear` Komponenten angezeigt (im Gegensatz zu undefinierten `endyear` Komponenten). Siehe § 2.3.8 und Tabelle 2 auf Seite 41 für weitere Beispiele.

`day` field (integer)

Dieses Feld beinhaltet die Tageskomponente von dem `date`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Tageskomponente vom Start des Datums.

`month` field (integer)

Dieses Feld ist der `month`, wie er in der Datenbankdatei gegeben ist, oder es beinhaltet die Monatskomponente vom `date`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Monatskomponente vom Start des Datums.

`year` field (integer)

Dieses Feld ist das `year`, wie es in der Datenbankdatei gegeben ist, oder es beinhaltet die Jahreskomponente vom `date`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Jahreskomponente vom Start des Datums.

| <b>bib-Datei</b> |                        | <b>Daten-Interface</b> |                        |
|------------------|------------------------|------------------------|------------------------|
| <b>Feld</b>      | <b>Wert (Beispiel)</b> | <b>Feld</b>            | <b>Wert (Beispiel)</b> |
| date             | 1988                   | day                    | undefined              |
|                  |                        | month                  | undefined              |
|                  |                        | year                   | 1988                   |
|                  |                        | endday                 | undefined              |
|                  |                        | endmonth               | undefined              |
|                  |                        | endyear                | undefined              |
| date             | 1997/                  | day                    | undefined              |
|                  |                        | month                  | undefined              |
|                  |                        | year                   | 1997                   |
|                  |                        | endday                 | undefined              |
|                  |                        | endmonth               | undefined              |
|                  |                        | endyear                | empty                  |
| urldate          | 2009-01-31             | urlday                 | 31                     |
|                  |                        | urlmonth               | 01                     |
|                  |                        | urlyear                | 2009                   |
|                  |                        | urlendday              | undefined              |
|                  |                        | urlendmonth            | undefined              |
|                  |                        | urlendyear             | undefined              |
| origdate         | 2002-01/2002-02        | origday                | undefined              |
|                  |                        | origmonth              | 01                     |
|                  |                        | origyear               | 2002                   |
|                  |                        | origendday             | undefined              |
|                  |                        | origendmonth           | 02                     |
|                  |                        | origendyear            | 2002                   |
| eventdate        | 1995-01-31/1995-02-05  | eventday               | 31                     |
|                  |                        | eventmonth             | 01                     |
|                  |                        | eventyear              | 1995                   |
|                  |                        | eventendday            | 05                     |
|                  |                        | eventendmonth          | 02                     |
|                  |                        | eventendyear           | 1995                   |

Tabelle 7: Daten-Interface

**endday** field (integer)

Falls die Datumsspezifikation in dem `date`-Feld ein Bereich ist, beinhaltet dieses Feld die Tageskomponente vom Ende des Datums.

**endmonth** field (integer)

Falls die Datumsspezifikation in dem `date`-Feld ein Bereich ist, beinhaltet dieses Feld die Monatskomponente vom Ende des Datums.

**endyear** field (integer)

Falls die Datumsspezifikation in dem `date`-Feld ein Bereich ist, beinhaltet dieses Feld die Jahreskomponente vom Ende des Datums. Eine leere (aber definierte) `endyear`-Komponente deutet auf einen offenen `date`-Bereich.

**origday** field (integer)

Dieses Feld beinhaltet die Tageskomponente vom `origdate`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Tageskomponente vom Start des Datums.

**origmonth** field (integer)

Dieses Feld beinhaltet die Monatskomponente vom `origdate`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Monatskomponente vom Start des Datums.

**origyear** field (integer)

Dieses Feld beinhaltet die Jahreskomponente vom `origdate`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Jahreskomponente vom Start des Datums.

**origendday** field (integer)

Falls die Datumsspezifikation in dem `origdate`-Feld ein Bereich ist, beinhaltet dieses Feld die Tageskomponente vom Ende des Datums.

**origendmonth** field (integer)

Falls die Datumsspezifikation in dem `origdate`-Feld ein Bereich ist, beinhaltet dieses Feld die Monatskomponente vom Ende des Datums.

**origendyear** field (integer)

Falls die Datumsspezifikation in dem `origdate`-Feld ein Bereich ist, beinhaltet dieses Feld die Jahreskomponente vom Ende des Datums. Eine leere (aber definierte) `origendyear`-Komponente deutet auf einen offenen `origdate`-Bereich.

`eventday` field (integer)

Dieses Feld beinhaltet die Tageskomponente vom `eventdate`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Tageskomponente vom Start des Datums.

`eventmonth` field (integer)

Dieses Feld beinhaltet die Monatskomponente vom `eventdate`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Monatskomponente vom Start des Datums.

`eventyear` field (integer)

Dieses Feld beinhaltet die Jahreskomponente vom `eventdate`-Feld. Falls das Datum ein Bereich ist, beinhaltet es die Jahreskomponente vom Start des Datums.

`eventendday` field (integer)

Falls die Datumsspezifikation in dem `eventdate`-Feld ein Bereich ist, beinhaltet dieses Feld die Tageskomponente vom Ende des Datums.

`eventendmonth` field (integer)

Falls die Datumsspezifikation in dem `eventdate`-Feld ein Bereich ist, beinhaltet dieses Feld die Monatskomponente vom Ende des Datums.

`eventendyear` field (integer)

Falls die Datumsspezifikation in dem `eventdate`-Feld ein Bereich ist, beinhaltet dieses Feld die Jahreskomponente vom Ende des Datums. Eine leere (aber definierte) `eventendyear`-Komponente deutet auf einen offenen `eventdate`-Bereich.

`urlday` field (integer)

Dieses Feld beinhaltet die Tageskomponente vom `urldate`-Feld.

`urlmonth` field (integer)

Dieses Feld beinhaltet die Monatskomponente vom `urldate`-Feld.

`urlyear` field (integer)

Dieses Feld beinhaltet die Jahreskomponente vom `urldate`-Feld.

`urlendday` field (integer)

Falls die Datumsspezifikation in dem `urldate`-Feld ein Bereich ist, beinhaltet dieses Feld die Tageskomponente vom Ende des Datums.

`urlendmonth` field (integer)

Falls die Datumsspezifikation in dem `urldate`-Feld ein Bereich ist, beinhaltet dieses Feld die Monatskomponente vom Ende des Datums.

`urlendyear` field (integer)

Falls die Datumsspezifikation in dem `urldate`-Feld ein Bereich ist, beinhaltet dieses Feld die Jahreskomponente vom Ende des Datums. Eine leere (aber definierte) `urlendyear`-Komponente deutet auf einen offenen `urldate`-Bereich.

### 4.3. Zitatstile

Ein Zitatstil ist ein Set von Kommandos, wie `\cite`, welches verschiedene Typen von Zitaten druckt. Solche Stile sind in den Daten mit dem Suffix `cbx` definiert. Das `biblatex`-Paket lädt die ausgewählte Zitatstildatei am Ende des Paketes. Beachten Sie, dass ein kleines Repertoire von oft verwendeten Makros, mehrerer Standardzitatstile auch in `biblatex.def` enthalten sind. Diese Datei wird ebenso am Ende des Paketes, vor dem ausgewählten Zitatstil geladen. Es enthält auch die Definition der Befehle von § 3.6.5.

#### 4.3.1. Zitatstildaten

Bevor wir zu den individuell verfügbaren Befehlen in der Zitatstildatei übergehen, betrachten Sie dieses Beispiel der Gesamtstruktur einer typischen `cbx` Datei:

```
\ProvidesFile{example.cbx}[2006/03/15 v1.0 biblatex citation style]

\DeclareCiteCommand{\cite}{...}{...}{...}{...}
\DeclareCiteCommand{\parencite}[\mkbibparens]{...}{...}{...}{...}
\DeclareCiteCommand{\footcite}[\mkbibfootnote]{...}{...}{...}{...}
\DeclareCiteCommand{\textcite}{...}{...}{...}{...} \endinput
```

`\RequireCitationStyle{<style>}`

Dieser Befehl ist optional und für spezialisierte Zitatstile gedacht, die auf einem allgemeineren Stil aufbauen. Es lädt den Zitatstil `style.cbx`.

`\InitializeCitationStyle{<code>}`

Spezifiziert einen willkürlichen `<code>`, der erforderlich ist, um den Zitatstil zu initialisieren oder neu zu erfassen. Dieser Haken wird einmal bei der Paketladezeit durchgeführt und jedes Mal, wenn der Befehl `\citereset` von § 3.6.8 verwendet wird. Der Befehl `\citereset` erfasst ebenso die inneren Zitattracker (Spurenleser) dieses Pakets neu. Das Zurücksetzen (Neuerfassen) betrifft `\ifciteseen`, `\ifentryseen`, `\ifcitebid` und `\ifciteidem` Untersuchungen, welche in § 4.6.2 besprochen wurden. Wenn es in einer `refsection`-Umgebung benutzt wird, dann ist die Zurücksetzung der Zitattracker lokal zur gegenwärtigen `refsection`-Umgebung.

`\OnManualCitation{<code>}`

Spezifiziert einen willkürlichen `<code>`, der für ein teilweises Zurücksetzen des Zitatstils erforderlich ist. Dieser Haken wird jedes Mal durchgeführt, wenn der Befehl `\mancite`, von § 3.6.8 verwendet wird. Es ist in Zitatstilen besonders nützlich, die wiederholten Zitate durch Abkürzungen wie ‚ibidem‘ or ‚op. cit.‘ zu ersetzen, welche zweideutig sein können, wenn sie automatisch erzeugt werden und manuelle Zitate werden so gemischt. Der Befehl `\mancite` setzt ebenso die inneren ‚ibidem‘ und ‚idem‘ Tracker dieses Pakets zurück. Das Zurücksetzen betrifft die `\ifciteibid` und `\ifciteidem` Untersuchung, welche in § 4.6.2. besprochen wurden.

`\DeclareCiteCommand{<command>}[<wrapper>]{<precode>}{<loopcode>}{<sepcode>}{<postcode>}`

`\DeclareCiteCommand*{<command>}[<wrapper>]{<precode>}{<loopcode>}{<sepcode>}{<postcode>}`

Dies sind die Kernbefehle, welche benutzt werden, um alle Zitatbefehle zu bestimmen. Es gibt ein optionales und fünf obligatorische Argumente. Das `<comand>` ist der Befehl, um beispielsweise `\cite` zu definieren. Wenn das optionale Argument `<wrapper>` gegeben ist, wird das komplette Zitat als Argument zu diesem `<wrapper>` weitergegeben, in dem Fall muss der wrapper-Befehl ein obligatorisches Argument nehmen.<sup>27</sup> Der `<precode>` ist ein willkürlicher Code, der am Beginn des Zitates durchgeführt wird. Er bearbeitet normalerweise das `<prenote>`-Argument, welches im `prenote`-Datenfeld verfügbar ist. Es kann ebenso verwendet werden, um Makros zu initialisieren, die vom `<loopcode>` gefordert werden. Der `<loopcode>` ist ein willkürlicher Code, um jeden, zum `<command>` passenden Zugangsschlüssel durchzuführen. Das ist der Kerncode, welcher die Zitatbezeichnung oder jegliche anderen Daten druckt. Der `<sepcode>` ist ein willkürlicher Code, welcher nach jeder Wiederholung des `<loopcode>` durchgeführt wird. Er wird nur durchgeführt, wenn eine Liste von Zugangsschlüsseln an den `<command>` weitergegeben wird. Der `<sepcode>` fügt für gewöhnlich eine Art Aufteilung, wie ein Komma oder Semikolon ein. Der `<postcode>` ist ein willkürlicher Code welcher am Ende des Zitates durchgeführt wird. Der `<postcode>` handhabt typischerweise das `<postnote>`-Argument, welches im

<sup>27</sup>Typische Beispiele für wrapper-Befehle sind `\mkbibparens` und `mkbibfootnote`.

postnote-Feld verfügbar ist.<sup>28</sup> Die feste Variante von `\DeclareCiteCommand` definiert ein festes `<command>`. Zum Beispiel, `\DeclareCiteCommand*{cite}` würde `\cite*` definieren.<sup>29</sup>

`\DeclareMultiCiteCommand{<command>}[<wrapper>]{<cite>}{<delimiter>}`

Dieser Befehl definiert ‚multicite‘-Befehle (§ 3.6.3). Der `<command>` ist der mehrfach zitierte Befehl, zum Beispiel `\cites`, der definiert ist. Er ist automatisch robust gemacht. Mehrfach zitierende Befehle sind oben in Backendbefehlen definiert, mit `\DeclareCiteCommand` aufgebaut und das `<cite>` Argument gibt den Namen des Backendbefehls an, um benutzt zu werden. Beachten Sie, dass der Wrapper vom Backendbefehl (in dem Fall die Weiterleitung des `<wrapper>`-Arguments zu `\DeclareCiteCommand`) ignoriert wird. Benutzen Sie das optionale `<wrapper>`-Argument, um ein alternatives Wrapper anzugeben. Der `<delimiter>` ist der Strang, der als ein Teiler zwischen den individuellen Zitaten in der Liste gedruckt wird. Das ist typisch bei `\multicitedelim`. Die folgenden Beispiele sind reale Definitionen aus `biblatex.def`:

```
\DeclareMultiCiteCommand{\cites}%
{\cite}{\multicitedelim} \DeclareMultiCiteCommand{\parencites}[\mkbibparens]%
{\parencite}{\multicitedelim}
\DeclareMultiCiteCommand{\footcites}[\mkbibfootnote]%
{\footcite}{\multicitedelim}
```

`\DeclareAutoCiteCommand{<name>}[<position>]{<cite>}{<multicite>}`

Dieser Befehl bestimmt Definitionen für die Befehle `\autocite` und `\autocites`, von § 3.6.4. Die Definitionen werden mit der `autocite`-Paketoption von § 3.1.2.1 ermöglicht. Der `<name>` ist ein Bezeichner, welcher dazu dient, als Wert zur Paketoption weitergegeben zu werden. Die autozitierenden Befehle sind oben in Backendbefehlen, wie `\parencite` und `\parencites` aufgebaut. Die Argumente `<cite>` und `<multicite>` geben den Backendbefehl zur Benutzung an. Das Argument `<cite>` bezieht sich auf `\autocite` und `<multicite>` bezieht sich auf `\autocites`. Das Argument `<position>` kontrolliert die Handhabung von jeglichen Satzzeichen nach dem Zitat.

<sup>28</sup>Die für den `<loopcode>` verfügbaren bibliographische Datei, ist die Eintragsdatei, welche momentan durchläuft. Zusammengefasst heißt das, dass die Datei des ersten Eintrags für den `<precode>` verfügbar ist und die Datei des letzten für den `<precode>` verfügbar ist. ‚Ersterer‘ und ‚Letzterer‘ beziehen sich auf die Ordnung, in der die Zitate ausgedruckt werden. Wenn die `sortcites`-Paketauswahl aktiv ist, ist die Ordnung der Liste im Anschluss sortiert. Beachten Sie, dass keine Bibliographie-Datei für den `<sepcode>` verfügbar ist.

<sup>29</sup> Beachten Sie, dass die reguläre Variante von `command` eine feste Variante implizit definiert, es sei denn, die feste Variante wurde schon vorher definiert. Dies ist als ein Rückgriff beabsichtigt. Die implizite Definition ist ein Synonym für die reguläre Variante.

Mögliche Werte sind `l`, `r`, `f`. `r` bedeutet, dass die Zeichensetzung rechts vom Zitat gesetzt wird, das heißt, sie wird nicht bewegt. `l` bedeutet, dass die Zeichensetzung nach dem Zitat sich an die linke Seite des Zitats bewegt. Das ist nützlich bei Zitaten in Fußnoten, da es das Fußnotenzeichen nach der Endzeichensetzung legt. Dieses Argument ist optional und Vorgegeben für `r`. Siehe `\DeclareAutoPunctuation` in § 4.7.5 und die `autopunct` Paketoption in § 3.1.2.1. Die folgenden Beispiele sind Realdefinitionen, die von `biblatex.def` stammen.

```
\DeclareAutoCiteCommand{plain}{\cite}{\cites}
\DeclareAutoCiteCommand{inline}{\parencite}{\parencites}
\DeclareAutoCiteCommand{footnote}[1]{\footcite}{\footcites}
\DeclareAutoCiteCommand{footnote}[f]{\smartcite}{\smartcites}
```

### 4.3.2. Spezielle Bereiche

Die folgenden Datenfelder werden von `biblatex` benutzt, um Zitatbefehle weiterzuleiten. Sie werden nicht in der `bib`-Datei verwendet, aber automatisch durch das Paket definiert. Aus der Perspektive des Zitatstils unterscheidet sie sich nicht von den Datenfeldern in einer `bib`-Datei. Siehe § 4.2.4.

`prenote` field (literal)

Das Argument `<prenote>` wird an einen Zitatbefehl weitergeleitet. Dieses Datenfeld ist spezifisch für Zitate und nicht abrufbar in der Bibliographie. Wenn das Argument `<prenote>` leer ist, dann ist das Datenfeld undefiniert.

`postnote` field (literal)

Das Argument `<postnote>` wird an ein Zitatbefehl weitergeleitet. Dieses Datenfeld ist spezifisch für Zitate und nicht abrufbar in der Bibliographie. Wenn das Argument `<postnote>` fehlt oder leer ist, dann ist das Datenfeld undefiniert.

## 4.4. Datenanschluss

Der Datenanschluss sind die Anlagen, welche benutzt werden, um alle bibliographischen Daten zu formatieren und zu drucken. Diese Anlagen sind abrufbar in beiden Bibliographie- und Zitatstilen.

### 4.4.1. Dateibefehle

Dieses Datenfeld stellt den Hauptdatenanschluss vom `biblatex` Paket vor. Dies sind die Befehle, welche die meiste Arbeit verrichten, d. h. sie drucken eigentlich die Datei, welche die Listen und Datenfelder voraussetzen.

`\printfield[⟨format⟩]{⟨field⟩}`

Dieser Befehl druckt ein *⟨field⟩*, welches die formatierende Anordnung *⟨format⟩*, als `\DeclareFieldFormat` benutzt. Wenn ein typspezifiziertes *⟨format⟩* deklariert ist, dann hat die typspezifiziert formatierte Anordnung Vorrang vor der allgemeinen. Wenn das *⟨field⟩* undefiniert ist, wird nichts gedruckt. Wenn das *⟨format⟩* entfällt, dann benutzt `\printfield` den Namen des Datenfeldes als einen Formatnamen. Zum Beispiel, wenn das `title`-Datenfeld gedruckt wird und das *⟨format⟩* nicht spezifiziert ist, dann wird es versuchen das Bereichsformat `title` zu nutzen.<sup>30</sup> In diesem Fall wird ebenso eine beliebig typspezifizierte formatierte Anordnung Vorrang vor der allgemeinen haben. Falls all diese Formate undefiniert sind, so fallen sie als letzte Rettung in `default` (Standardeinstellung) zurück. Beachten Sie, dass `\printfield` den Namen des Datenfeldes liefert, welcher gegenwärtig in `\currentfield` behandelt wird, um in einer Feld-formatierenden Anordnung benutzt zu werden.

`\printlist[⟨format⟩][⟨start⟩–⟨stop⟩]{⟨literal list⟩}`

Dieser Befehl legt sich über alle Begriffe in einer *⟨literal list⟩*, beginnend von der Begriffsnummer *⟨start⟩* und endend bei Begriffsnummer *⟨stop⟩*, inklusive *⟨start⟩* und *⟨stop⟩* (alle Listennummern starten bei I). Jeder gedruckte Begriff benutzt die formatierende Anordnung *⟨format⟩*, als mit `\DeclareListFormat` definiert. Wenn ein typspezifiziertes *⟨format⟩* deklariert ist, dann hat die typspezifiziert formatierte Anordnung Vorrang vor der allgemeinen. Wenn das *⟨literal field⟩* undefiniert ist, wird nichts gedruckt. Wenn das *⟨format⟩* entfällt, dann benutzt `\printlist` den Namen des Bereichs als einen Formatnamen. In diesem Fall wird ebenso eine beliebig typspezifizierte formatierte Anordnung Vorrang vor der allgemeinen haben. Falls all diese Formate undefiniert sind, so fallen sie, als letzte Rettung in `default` (Standardeinstellung) zurück. Das *⟨start⟩*-Argument fällt für I weg; *⟨stop⟩* fällt für die gesamte Anzahl der Begriffe in der Liste weg. Wenn die Gesamtzahl größer ist als *⟨maxitems⟩*, fällt *⟨stop⟩* für *⟨minitems⟩* (siehe § 3.1.2.1). Siehe `\printnames` für weitere Details. Berücksichtigen Sie, `\printlist` liefert den Namen der Wortliste, welche gegenwärtig in `\currentlist` behandelt wird, um in einer Listen-formatierenden Anordnung benutzt zu werden.

`\printnames[⟨format⟩][⟨start⟩–⟨stop⟩]{⟨name list⟩}`

Dieser Befehl legt sich über alle Begriffe in einer *⟨name list⟩*, beginnend von Begriffsnummer *⟨start⟩* und endend bei Begriffsnummer *⟨stop⟩*, inklusive *⟨start⟩* und *stop* (alle Listennummern starten bei I). Jeder gedruckte Begriff benutzt die formatierende Anordnung *⟨format⟩* als mit `\DeclareNameFormat` definiert. Wenn ein

<sup>30</sup>Mit anderen Worten, `\printfield{title}` ist äquivalent zu `\printfield[title]{title}`.

typspezifiziertes *format* deklariert ist, dann hat die typspezifiziert formatierte Anordnung Vorrang vor der allgemeinen. Wenn das *name list* undefiniert ist, wird nichts gedruckt. Wenn das *format* entfällt, dann benutzt `\printlist` den Namen des Datenfeldes als einen Formatnamen. In diesem Fall wird ebenso eine beliebig typspezifizierte formatierte Anordnung Vorrang vor der allgemeinen haben. Falls all diese Formate undefiniert sind, so fallen sie, als letzte Rettung in `default` zurück. Das *start*-Argument fällt für I weg; *stop* fällt für die gesamte Anzahl der Begriffe in der Liste weg. Wenn die Gesamtzahl größer ist als *maxnames*, fällt *stop* für *minnames* aus (siehe § 3.1.2.1). Wenn Sie einen Bereich trennen wollen aber das Ausfall-Listen-Format benutzen wollen, dann muss das erste optionale Argument noch gegeben werden, ist aber links leer.

```
\printnames [] [1-3]{...}
```

Eines von *start* und *stop* sollte ausgelassen werden, weswegen die folgenden Argumente gültig sind:

```
\printnames [...] [-1]{...}
\printnames [...] [2-]{...}
\printnames [...] [1-3]{...}
```

Wenn Sie *maxnames* und *minnames* ausschalten und den Druck der Gesamtliste verhindern wollen, dann können Sie sich auf das `listtotal`-Verzeichnis im zweiten optionalen Argument beziehen.

```
\printnames [...] [-\value{listtotal}]{...}
```

Wann immer `\printnames` und `\printlist` eine Liste ausführen, ist eine Information bezüglich der aktuellen Lage über den Weg von vier Verzeichnissen zugänglich: Das `listtotal`-Verzeichnis hält die gesamte Anzahl von Begriffen in der aktuellen Liste fest, `listcount` hält die Anzahl der momentan bearbeiteten Begriffe fest, `liststart` ist das *start*-Argument, das zu `\printnames` oder `\printlist` weitergeht. `liststop` ist das *stop* Argument. Diese Verzeichnisse sind dazu gedacht, in formatierenden Anweisungen benutzen zu werden. `listtotal` sollte vielleicht ebenso in einem zweiten optionalen Argument für `\printnames` und `\printlist` verwendet werden. Beachten Sie, dass diese Verzeichnisse für die formatierenden Anordnungen lokal sind und keine bedeutsamen Werte festhalten, wenn sie irgendwo anders verwendet werden. Für jede Liste gibt es ein Verzeichnis mit dem selben Namen, welches die gesamte Anzahl von Begriffen in der entsprechenden Liste enthält. Zum Beispiel enthält das `author`-Verzeichnis die gesamte Anzahl von Begriffen der `author`-Liste. Diese Verzeichnisse sind ähnlich wie `listtotal`, mit der Ausnahme, dass sie vielleicht ebenso unabhängig in der formatierenden Aussage-Liste verwendet werden. Dort gibt es ebenso `maxnames` und `minnames`, genauso wie `maxitems`

und minitems Verzeichnisse, welche die Werte der entsprechenden Paketooptionen enthalten. Siehe § 4.10.5 für eine komplette Liste solcher inneren Verzeichnisse. Berücksichtigen Sie `printnames`, dann liefert es den Namen auf der Namensliste, der momentan in `currentname` bearbeitet wird, um im formatierten Aussage-Namen verwendet zu werden.

`\printtext` [*format*] {*text*}

Dieser Befehl druckt *text*, welcher eventuell ein druckbarer Text oder ein beliebig kodegenerierter, druckbarer Text ist. Es verdeutlicht dem Zeichensetzungspuffer, bevor *text* eingebracht und `biblatex` informiert sind, dass druckbare Texte beigelegt sind. Das garantiert, dass alle vorangegangenen und folgenden `\newblock` und `\newunit` Befehle den gewünschten Effekt haben. `\printfield` und `\printnames` sowie `\bibstring` und seine Begleitbefehle (siehe § 4.8) tun dies automatisch. Die Benutzung dieses Befehls wird benötigt, wenn ein Bibliographiestil einen wörtlichen Text enthält (inklusive der Befehle von §§ 4.7.3 and 4.7.4), um zu gewährleisten, dass Block und Einheit der Zeichensetzung, wie in § 4.7.1 angekündigt, funktionieren. Das optionale *format*-Argument spezifiziert ein formatierendes Argumentfeld, um für das Format *format* verwendet werden zu können. Dies könnte nützlich sein, wenn einzelne Datenfelder in einem Stück gedruckt werden sollen, z. B. in Klammern oder Anführungszeichen eingeschlossenen Gesamtstücken. Wenn ein typspezifiziertes *format* angegeben ist, dann hat die typspezifizierende Formatierung direkten Vorrang, vor dem Allgemeinen einen. Wenn das *format* ausgelassen wird, dann wird der *text* so gedruckt, wie er ist. Siehe ebenso in § 4.11.7 für einige praktische Hinweise.

`\printfile` [*format*] {*file*}

Dieser Befehl ist ähnlich wie `\printtext`, mit der Ausnahme, dass das zweite Argument ein Dateiname anstatt eines wörtlichen Textes ist. Das Argument *file* muss der Name einer gültigen LaTeX-Datei sein, die in TeX-Suchpfad gefunden wird. `\printfile` verwendet `\input`, um *file* zu laden. Wenn dort keine jener Dateien ist, dann tut `\printfile` nichts. Das optionale *format*-Argument spezifiziert ein bereichformatierendes Argument, um für *file* angewendet zu werden. Wenn ein typspezifizierendes *format* angegeben ist, dann hat das typspezifizierende formatierende Argument Vorrang vor dem allgemeinen. Wenn das *format* ausgelassen wird, dann wird *file* so gedruckt, wie es ist. Beachten Sie, dass dieser Zusatz ausführlich das Setzen der Paketooption `loadfiles` von § 3.1.2.1 ermöglicht. Bei Verzug wird `\printfile` keine einzige Datei einlesen.

`\printdate` Dieser Befehl druckt das Eingabedatum als spezialisiert in den Datenfeldern `date` oder `month/year` ein. Das Format wird durch die Paketooption `date`, von § 3.1.2.1, kontrolliert. Beachten Sie, dass dieser Befehl eine Nahtstelle mit dem Zeichenset-

zungsspurenleser ist. Es gibt keinen Grund, ihn in ein `\printtext` Befehl zu verpacken.

`\printdateextra` Es ist ähnlich, wie `\printdate`, gliedert aber das `extrayear`-Datenfeld in die Datumsspezifikation mit ein. Dies ist wertvoll für Autor-Jahr Zitat bestimmte Bibliografistile.

`\printurldate` Dieser Befehl druckt das `urldate` der Eingabe. Das Format wird durch die Paketoption `urldate`, von § 3.1.2.1, kontrolliert. Beachten Sie, dass dieser Befehl eine Nahtstelle mit dem Zeichensetzungsspurnleser ist. Es gibt keinen Grund, es in einen `\printtext`-Befehl zu verpacken.

`\printorigdate` Dieser Befehl druckt das `origdate` der Eingabe. Das Format wird durch die Paketoption `origdate`, von § 3.1.2.1, kontrolliert. Beachten Sie, dass dieser Befehl eine Nahtstelle mit dem Zeichensetzungsspurnleser ist. Es gibt keinen Grund, ihn in ein `\printtext`-Befehl zu verpacken.

`\printeventdate` Dieser Befehl druckt das `eventdate` der Eingabe. Das Format wird durch die Paketoption `eventdate`, von § 3.1.2.1, kontrolliert. Beachten Sie, dass dieser Befehl eine Nahtstelle mit dem Zeichensetzungsspurnleser ist. Es gibt keinen Grund, ihn in ein `\printtext` Befehl zu verpacken.

`\indexfield`[*format*]{*field*}

Dieser Befehl ist gleichzusetzen mit `\printfield`, mit der Ausnahme, dass *field* nicht gedruckt wird, aber in die Indexbenutzung, mittels der formatierenden Anweisung *format*, die mit `\DeclareIndexFieldFormat` definiert sind, hinzu gefügt werden. Wenn ein Typspezifizierungs-*format* angegeben ist, dann hat es Vorrang vor den allgemeinen. Wenn *field* undefiniert ist, dann tut dieser Befehl nichts. Wird das *format* ausgelassen, dann versucht `\indexfield` den Namen des Datenfeldes als einen Formatnamen zu verwenden. In diesem Falle wird irgendeine typspezifizierende Formatanordnung den Vorrang vor den allgemeinen einnehmen. Wenn all die Formate undefiniert sind, so fallen sie zurück zu `default`, als ein letzter Ausweg.

`\indexlist`[*format*][*start*–*stop*]{*literal list*}

Dieser Befehl ist gleichzusetzen mit `\printlist`, mit der Ausnahme, dass die Begriffe nicht gedruckt werden, aber in die Indexbenutzung, mittels der formatierenden Anweisung *format*, die mit `\DeclareIndexListFormat` definiert sind, hinzugefügt werden. Wenn ein Typspezifizierungs- *format* angegeben ist, dann hat die typspezialisierende Formatierendeanordnung Vorrang vor den allgemeinen. Wenn *literal list* undefiniert ist, dann tut dieser Befehl nichts. Wird das *format* ausgelassen, dann versucht `\indexlist` den Namen des Feldes als einen Formatnamen zu verwenden. In diesem Falle wird irgendeine typspezifizierende Formatanordnung

den Vorrang vor den allgemeinen einnehmen. Wenn all die Formate undefiniert sind, so fallen sie zurück zu `default`, als ein letzter Ausweg.

`\indexnames` [*format*] [*start*–*stop*] {*name list*}

Dieser Befehl ist gleichzusetzen mit `\printnames`, mit der Ausnahme, dass die Begriffe nicht gedruckt werden, aber in die Indexbenutzung, mittels der formatierenden Anweisung *format*, die mit `\DeclareIndexNameFormat` definiert sind, hinzu gefügt werden. Wenn ein Typspezifizierung *format* angegeben ist, dann hat die typspezialisierende Formatierendeanordnung Vorrang vor den allgemeinen. Wenn *name list* undefiniert ist, dann tut dieser Befehl nichts. Wird das *format* ausgelassen, dann versucht `\indexname` den Namen des Datenfeldes als einen Formatnamen zu verwenden. In diesem Falle wird irgendeine typspezifizierende Formatanordnung den Vorrang vor den allgemeinen einnehmen. Wenn all die Formate undefiniert sind, so fallen sie zurück zu `default`, als ein letzter Ausweg.

`\entrydata`{*key*}{*code*}

`\entrydata*`{*key*}{*code*}

Dateibefehle, wie `\printfield`, benutzen normalerweise die Datei der Eingabe, die momentan bearbeitet wird. Sie sollten eventuell `\entrydata` benutzen, um Zusammenhänge lokal umzutauschen. Der *key* ist der Zugangsschlüssel, der lokal zu verwendenden Eingabe. Der *code* ist ein beliebiger Kode, um in diesem Kontext ausgeführt zu werden. Dieser Kode wird in einer Gruppe ausgeführt. Siehe § 4.11.6 als Beispiel. Beachten Sie, dass dieser Befehl automatisch die Sprachen verändert, wenn die `babel`-Paketoption aktiv ist. Die feste Version `\entrydata*` vervielfältigt alle Datenfeld des eingeschlossenen Eintrags, benutzt Bereich, Verzeichnis und andere Quellennamen, die mit dem Präfix `,saved` verkettet sind. Dies ist wichtig, wenn man zwei Datensets vergleichen will. Zum Beispiel im *code*-Argument, wo der `author`-Bereich den Autor der Eingabe *key* enthält und der eingeschlossenen Autor der Eingabe verfügbar ist als `savedauthor`. Das `author`-Verzeichniss enthält die Anzahl der Namen im `author`-Datenfeld des *key*; das `savedauthor`-Verzeichnis referiert auf das Autorenverzeichnis der eingeschlossenen Eingabe.

`\entryset`{*precode*}{*postcode*}

Dieser Befehl ist bestimmt, um in der Bibliographie als `@set` Eingabe anders benutzt zu werden. Es schlingt sich, als `entryset` bezeichnet, über alle Mitglieder des Satzes und führt den passenden Treiber für das jeweilige Satz-Mitglied durch. Das ist der Durchführung des `\usedriver`-Befehls, für jedes Satzmitglied, aus § 4.6.4, ähnlich. Der *precode* ist ein beliebiger Kode, der vor der Verarbeitung jedes Artikels im Satz durchzuführen ist. Der *postcode* ist ein beliebiger Kode, der sofort zur Verarbeitung jedes Artikels durchzuführen ist. Beide Argumente sind verbindlich in Bezug auf die Syntax, können aber leer gelassen werden. Siehe § 4.11.1 für Gebrauchsbeispiele.

#### 4.4.2. Formatieren von Anordnungen

Dieser Abschnitt führt die Befehle ein, die dazu verwendet werden, die Formatierungsanordnungen, die von den Dateibefehlen von § 4.4.1 benötigt werden, zu definieren. Beachten Sie, dass alle Standardformate in `biblatex.def` definiert sind.

```
\DeclareFieldFormat[⟨entrytype, ...⟩]{⟨format⟩}{⟨code⟩}
```

```
\DeclareFieldFormat*{⟨format⟩}{⟨code⟩}
```

Diese Befehle definieren das Bereichsformat `⟨format⟩`. Diese Formatierungsanordnung ist ein beliebiger `⟨code⟩`, um von `\printfield` ausgeführt zu werden. Der Wert des Bereichs wird an den `⟨code⟩` weitergegeben als das erste und einzige Argument. Der Name des Bereichs, der gegenwärtig bearbeitet wird, ist für den `⟨code⟩` als `\currentfield` verfügbar. Wenn ein `⟨entrytype⟩` spezifiziert ist, dann ist das Format für diesen Typ spezifiziert. Das Argument `⟨entrytype⟩` kann eine Komma-getrennte Liste von Werten sein. Die feste Variante dieses Befehls ist der regelmäßigen Version ähnlich, außer dass alle mit dem Typ spezifischen Formate gelöscht werden.

```
\DeclareListFormat[⟨entrytype, ...⟩]{⟨format⟩}{⟨code⟩}
```

```
\DeclareListFormat*{⟨format⟩}{⟨code⟩}
```

Definieren das Bereichsformat `⟨format⟩`. Diese Formatierungsanordnung ist ein beliebiger `⟨code⟩`, um von `\printlist` ausgeführt zu werden. Der gegenwärtige Begriff wird an den `⟨code⟩` weitergegeben als das erste und einzige Argument. Der Name der wörtlichen Liste, die gegenwärtig bearbeitet wird, ist für den `⟨code⟩` als `\currentlist` verfügbar. Wenn ein `⟨entrytype⟩` spezifiziert ist, dann ist das Format für diesen Typ spezifiziert. Das Argument `⟨entrytype⟩` kann eine Komma-getrennte Liste von Werten sein. Beachten Sie, dass die Formatierungsanordnung auch die Zeichensetzung behandelt, die zwischen den individuellen Begriffen in der Liste einzufügen sind. Sie müssen überprüfen, ob Sie in der Mitte oder am Ende der Liste sind, in dem Falle, ob `listcount` kleiner oder gleich ist mit `liststop`. Die Sternchenvariante dieses Befehls ist der regelmäßigen Version ähnlich, außer dass alle typspezifischen Formate gelöscht sind.

```
\DeclareNameFormat[⟨entrytype, ...⟩]{⟨format⟩}{⟨code⟩}
```

```
\DeclareNameFormat*{⟨format⟩}{⟨code⟩}
```

Definieren das Bereichsformat `⟨format⟩`. Diese Formatierungsanordnung ist ein beliebiger `⟨code⟩`, um von `\printnames` ausgeführt zu werden. Wenn ein `⟨entrytype⟩` spezifiziert ist, dann ist das Format zu diesem Typ spezifisch. Das Argument `⟨entrytype⟩` kann eine Komma-getrennte Liste von Werten sein. Die individuellen Teile eines Namens werden an den `⟨code⟩`, als separates Argument weitergegeben. Diese Argumente sind wie folgt:

- #1 Die Nachnamen. Wenn ein Name nur aus einem Teil besteht (zum Beispiel, ‚Aristotle‘), dann wird dieser Teil als Nachname behandelt.
- #2 Der Nachname, mit Initialien angegeben.
- #3 Der Vorname. Dieses Argument beinhaltet auch alle Mittelnamen.
- #4 Der Vorname, mit Initialien angegeben.
- #5 Die Namenspräfixe, wie zum Beispiel von, van, of, da, de, del, della, etc. Beachten Sie, dass Namenspräfixe referierend sind, wie der ‚von part‘ des Namens in der BibTeX-Dokumentation.
- #6 Die Namenspräfixe, mit Initialien angegeben.
- #7 Die Namensaffixe, wie zum Beispiel ‚junior‘, ‚senior‘, ‚der Jüngere‘, ‚der Ältere‘, etc. Bedenken sie, dass Namensaffixe referierend, wie der ‚junior part‘ des Namens in der BibTeX-Dokumentation.
- #8 Die Namensaffixe, mit Initialien angegeben.

Wenn ein bestimmter Teil eines Namens nicht verfügbar ist, dann wird das entsprechende Argument leer sein, folglich können Sie `\ifblank` Untersuchungen benutzen, um die individuellen Teile eines Namens zu überprüfen. Der Name auf der Namensliste, der gegenwärtig bearbeitet wird, ist für `<code>` als `\currentname` verfügbar. Alle Initialen sind empfindlich für die Paketooption, siehe § 3.11.4 für Details. Beachten Sie, die Formatierungsanordnung behandelt auch die Zeichensetzung, die zwischen getrennten Namen und zwischen den individuellen Teilen eines Namens einzufügen ist. Sie brauchen eine Kontrolle, ob Sie in der Mitte oder am Ende der Liste sind, in dem Fall ob `listcount` kleiner als oder gleich ist mit `liststop`. Die Sternchenvariante dieses Befehls ist der regelmäßigen Version ähnlich, außer dass alle typspezifischen Formate gelöscht sind.

```
\DeclareIndexFieldFormat [<entrytype, ...>] {<format>} {<code>}
\DeclareIndexFieldFormat* {<format>} {<code>}
```

Definieren das Bereichsformat `<format>`. Diese Formatierungsanordnung ist ein beliebiger `<code>`, um von `\indexfield` ausgeführt zu werden. Der Wert des Datenfeldes wird an den `<code>`, als das erste und einzige Argument weiter gegeben. Der Name des Bereichs, der gegenwärtig bearbeitet wird, ist für den `<code>` als `\currentfield` verfügbar. Wenn ein `<entrytype>` spezifiziert ist, dann ist das Format für diesen Typ spezifiziert. Das Argument `<entrytype>` kann eine Komma-getrennte Liste von Werten sein. Dieser Befehl ist identisch zu `\DeclareFieldFormat`, außer dass die durch den `<code>` behandelte Information nicht für den Druck beabsichtigt ist, aber in den Index geschrieben wird. Beachten Sie, dass `\indexfield` den `<code>` so durchführen wird, wie er ist, in dem Fall muss `<code>` oder ein ähnlicher Befehl in `\index` eingefügt werden. Die Sternchenvariante dieses Befehls ist der regelmäßigen Version ähnlich, außer dass alle typspezifischen Formate gelöscht sind.

```
\DeclareIndexListFormat [⟨entrytype, ... ⟩]{⟨format⟩}{⟨code⟩}
\DeclareIndexListFormat*{⟨format⟩}{⟨code⟩}
```

Definieren das Buchstabenlistenformat *⟨format⟩*. Diese Formatierungsanordnung ist ein beliebiger *⟨code⟩*, um für jeden Begriff in einer Liste, der in `\indexlist` bearbeitet wird, ausgeführt zu werden. Der gegenwärtige Begriff, wird an den *⟨code⟩*, als das erste und einzige Argument weiter gegeben. Der Name der wörtlichen Liste, die gegenwärtig bearbeitet wird, ist für den *⟨code⟩* als `\currentlist` verfügbar. Wenn ein *⟨entrytype⟩* spezifiziert ist, dann ist das Format für diesen Typ spezifiziert. Das Argument *⟨entrytype⟩* kann eine Komma-getrennte Liste von Werten sein. Dieser Befehl ist identisch zu `\DeclareListFormat`, außer dass die durch den *⟨code⟩* behandelte Information nicht für den Druck beabsichtigt ist, aber in den Index geschrieben wird. Beachten Sie, dass `\indexlist` den *⟨code⟩* so durchführen wird, wie er ist, in dem Fall muss *⟨code⟩* oder ein ähnlicher Befehl in `\index` eingefügt werden. Die Sternchenvariante dieses Befehls ist der regelmäßigen Version ähnlich, außer dass alle typspezifischen Formate gelöscht sind.

```
\DeclareIndexNameFormat [⟨entrytype, ... ⟩]{⟨format⟩}{⟨code⟩}
\DeclareIndexNameFormat*{⟨format⟩}{⟨code⟩}
```

Definieren die Namensliste *⟨format⟩*. Diese Formatierungsanordnung ist ein beliebiger *⟨code⟩*, um für jeden Namen in einer Liste, der in `\indexnames` bearbeitet wird, ausgeführt zu werden. Der Name der wörtlichen Liste, die gegenwärtig bearbeitet wird, ist für den *⟨code⟩* als `\currentname` verfügbar. Wenn ein *⟨entrytype⟩* spezifiziert ist, dann ist das Format für diesen Typ spezifiziert. Das Argument *⟨entrytype⟩* kann eine Komma-getrennte Liste von Werten sein. Die Teile des Namens werden an den *⟨code⟩*, als getrennte Argumente weitergegeben. Dieser Befehl ist sehr ähnlich zu `\DeclareNameFormat`, außer dass die durch den *⟨code⟩* behandelte Information nicht für den Druck beabsichtigt ist, aber in den Index geschrieben wird. Beachten sie, dass `\indexnames` den *⟨code⟩* so durchführen wird, wie er ist, in dem Fall muss *⟨code⟩* oder ein ähnlicher Befehl in `\index` eingefügt werden. Die Sternchenvariante dieses Befehls ist der regelmäßigen Version ähnlich, außer dass alle typspezifischen Formate gelöscht sind.

```
\DeclareFieldAlias [⟨entry type⟩]{⟨alias⟩}[⟨format entry type⟩]{⟨format⟩}
```

Erklärt *⟨alias⟩* zu einen Decknamen für das Bereichsformat *⟨format⟩*. Wenn ein *⟨entrytype⟩* spezifiziert ist, dann ist der Deckname für diesen Typ spezifiziert. Der *⟨format entry type⟩* ist der Eingabetyp des Backendformats. Dies ist nur erforderlich, wenn ein Deckname für eine typspezifizierende Formatierungsanordnung angegeben ist.

`\DeclareListAlias`[ $\langle entry type \rangle$ ]{ $\langle alias \rangle$ }[ $\langle format entry type \rangle$ ]{ $\langle format \rangle$ }

Erklärt  $\langle alias \rangle$  zu einem Decknamen für das Wortlistenformat  $\langle format \rangle$ . Wenn ein  $\langle entry type \rangle$  spezifiziert ist, dann ist der Deckname für diesen Typ spezifiziert. Der  $\langle format entry type \rangle$  ist der Eingabetyp des Backendformats. Dies ist nur erforderlich, wenn ein Deckname für eine typspezifizierende Formatierungsanordnung angegeben ist.

`\DeclareNameAlias`[ $\langle entry type \rangle$ ]{ $\langle alias \rangle$ }[ $\langle format entry type \rangle$ ]{ $\langle format \rangle$ }

Erklärt  $\langle alias \rangle$  zu einem Decknamen für das Namenslistenformat  $\langle format \rangle$ . Wenn ein  $\langle entry type \rangle$  spezifiziert ist, dann ist der Deckname für diesen Typ spezifiziert. Der  $\langle format entry type \rangle$  ist der Eingabetyp des Backendformats. Dies ist nur erforderlich, wenn ein Deckname für eine typspezifizierende Formatierungsanordnung angegeben ist.

`\DeclareIndexFieldAlias`[ $\langle entry type \rangle$ ]{ $\langle alias \rangle$ }[ $\langle format entry type \rangle$ ]{ $\langle format \rangle$ }

Erklärt  $\langle alias \rangle$  zu einem Decknamen für das Feldformat  $\langle format \rangle$ . Wenn ein  $\langle entry type \rangle$  spezifiziert ist, dann ist der Deckname für diesen Typ spezifiziert. Der  $\langle format entry type \rangle$  ist der Eingabetyp des Backendformats. Dies ist nur erforderlich, wenn ein Deckname für eine typspezifizierende Formatierungsanordnung angegeben ist.

`\DeclareIndexListAlias`[ $\langle entry type \rangle$ ]{ $\langle alias \rangle$ }[ $\langle format entry type \rangle$ ]{ $\langle format \rangle$ }

Erklärt  $\langle alias \rangle$  zu einem Decknamen für das Wortlistenformat  $\langle format \rangle$ . Wenn ein  $\langle entry type \rangle$  spezifiziert ist, dann ist der Deckname für diesen Typ spezifiziert. Der  $\langle format entry type \rangle$  ist der Eingabetyp des Backendformats. Dies ist nur erforderlich, wenn ein Deckname für eine typspezifizierende Formatierungsanordnung angegeben ist.

`\DeclareIndexNameAlias`[ $\langle entry type \rangle$ ]{ $\langle alias \rangle$ }[ $\langle format entry type \rangle$ ]{ $\langle format \rangle$ }

Erklärt  $\langle alias \rangle$  zu einem Decknamen für das Namenslistenformat  $\langle format \rangle$ . Wenn ein  $\langle entry type \rangle$  spezifiziert ist, dann ist der Deckname für diesen Typ spezifiziert. Der  $\langle format entry type \rangle$  ist der Eingabetyp des Backendformats. Dies ist nur erforderlich, wenn ein Deckname für eine typspezifizierende Formatierungsanordnung angegeben ist.

## 4.5. Anpassung

### 4.5.1. Aussortierung

Zusätzlich zu den vorher bestimmten Sortierungsentwürfen, in § 3.4 diskutiert, ist es möglich, neu zu definieren oder schlechte Definitionen zu modifizieren. Der Sortierungsprozess kann weiterhin durch das Ausschließen bestimmter Datenfelder angepasst werden, um auf pro-typ-Basis zu sortieren und den `presort`-Bereich auf pro-typ-Basis automatisch zu bestücken. Beachten Sie, dass die Anpassungspläne Biber verlangen. Diese werden mit keinem anderen Backend arbeiten.

`\DeclareSortingScheme{<name>}{<specification>}`

Biber only

Definiert den Sortierplan `<name>`. `<name>` ist der Identifikator, der an die `sorting`-Option (§ 3.1.2.1) weiter gegeben wird, wenn der Sortierplan ausgewählt wird. Die `<specification>` ist eine unbegrenzte Liste von `\sort`-Anordnungen, welche die Elemente spezifizieren, um in einem Sortierungsprozess berücksichtigt zu werden. Abstände, Anhänge und Linienenden können frei verwendet werden, um die `<specification>` visuell einzuordnen. Leere Linien sind nicht erlaubt. Dieser Befehl kann nur in der Präambel verwendet werden.

`\sort{<elements>}`

Gibt die im Sortierungsprozess betrachteten Elemente an. `<elements>` sind eine unbegrenzte Liste von `\name`, `\list`, `\field`, `\literal`, und `\citeorder` Befehlen, welche in der Ordnung bewertet werden, in der sie gegeben sind. Wenn ein Element definiert ist, dann wird es zum Gattungsschlüssel und das Sortierungsprogramm springt zur folgenden `\sort`-Anordnung. Wenn es undefiniert ist, dann wird das nächste Element bewertet. Da wörtliche Abläufe immer definiert werden, sollten irgendwelche `\literal`-Befehle das alleinige oder das letzte Element in einer `\sort` Anordnung sein. Der Befehl `\sort` unterstützt die folgenden optionalen Argumente:

`direction=true, false` default: ascending  
 ascending, descending

Die Sortierungsanordnung, die auch steigend oder fallend sein kann. Die Standardeinstellung ist eine steigende Anweisung.

`final=true, false` default: false

Diese Option kennzeichnet eine `\sort`-Anordnung als die in `<specification>` endgültige. Wenn eins der `<elements>` verfügbar ist, dann wird der Rest der `<specification>` ignoriert. Die kurze Form `final` ist äquivalent zu `final final=true`.

`sortcase=true, false`

Ungeachtet dessen, ob man argumentsensibel sortieren soll oder nicht. Die Standardeinstellung hängt von der umfassenden `sortcase`-Option ab.

`sortupper=true, false`

Ungeachtet dessen ob man 'uppercase before lowercase' (`true`) oder 'lowercase before uppercase'-Anweisung (`false`) sortieren soll oder nicht. Die Standardeinstellung hängt von der umfassenden Auswahl ab.

`\name{⟨name list⟩}`

Das Element `\name` fügt eine `⟨name list⟩` hinzu, um die Spezifikationen zu sortieren. Wenn die `⟨name list⟩` undefiniert ist, dann wird das Element ausgelassen.

`\list{⟨literal list⟩}`

Das Element `\list` fügt eine `⟨literal list⟩` hinzu, um die Spezifikationen zu sortieren. Wenn die `⟨literal list⟩` undefiniert ist, dann wird das Element ausgelassen.

`\field[⟨key=value, ...⟩]{⟨field⟩}`

Das Element `\field` fügt ein `⟨field⟩` hinzu, um die Spezifikationen zu sortieren. Wenn das `⟨field⟩` undefiniert ist, dann wird das Element ausgelassen.

`padding=left, right` default: left

Polstert einen Bereich auf der `left` oder `right` Seite aus, benutzt `padchar`, so dass seine Breite `padding` ist. Wenn keine Polsteroption gesetzt wird, wird überhaupt nicht gepolstert. Wenn irgendeine Polsteroption spezifiziert ist, dann wird das Polstern durchgeführt und die fehlenden Optionen werden beauftragt, in Standardwerte eingebaut zu werden. Polstern ist mit numerischen Feldern besonders nützlich. Zum Beispiel polstert der Befehl

```
\field[padding=left,padding=2,padchar=0]{volume}
```

den `volume`-Bereich mit Führungsnullen zu einer Breite von zwei Charakteren aus. Auf diese Weise werden Inhalte durch den numerischen Wert (01/02/11/12) aber nicht in der alphabetischen Ordnung (1/11/12/2) sortiert.

`padding=⟨integer⟩` default: 4

Die Zielbreite in Charakteren.

`padding=⟨character⟩` default: 0

Der Charakter wird benutzt, um das Feld auszupolstern.

`strside=left, right` default: left

Führt ein Teilkettenübereinstimmung auf der `left`-Seite des Bereiches durch. Die Zahl von Charakteren, zur übereinstimmung, wird durch die entsprechende `strwidth`-Option angegeben. Wenn keine Teilkettenoption gesetzt wird, wird überhaupt keine Teilkette, die zusammenpasst, durchgeführt. Wenn irgendeine Teilkettenoption

angegeben wird, dann wird die zusammenpassende Teilkette durchgeführt und die fehlenden Optionen werden eingebauten Verzugswerten zugeteilt. Wenn beide, Polsterung und Teilkettenübereinstimmung spezifiziert sind, dann wird die Teilkettenübereinstimmung zuerst aufgeführt.

`strwidth=<integer>` default: 4

Die Nummer der Charaktere, die zusammenpassen.

`\literal{<string>}`

Das Element `\literal` addiert ein wörtliches `<string>` mit der Sortierspezifikation. Das ist als Rückgriff nützlich, falls einiger Felder nicht verfügbar sind.

`\citeorder` Das Element `\citeorder` hat eine besondere Bedeutung. Es zeigt an, dass die Einträge in der Ordnung behalten werden sollten, in der sie zitiert wurden. Es ist gewöhnlich das einzige Element in der `<specification>`.

```
\DeclareSortingScheme{none}{
\sort{\citeorder}
}
```

Hier sind einige Beispiele. Im ersten Beispiel definieren wir einen einfachen Name/Titel/Jahr-Entwurf. Das Namenselement kann der `author`, der `editor` oder der `translator` sein. In Anbetracht dieser Spezifizierung wird das Sortierhilfsprogramm das erste Element verwenden, das verfügbar ist und es mit dem `title` weiterführen. Beachten Sie, dass die Optionen `useauthor`, `useeditor` und `usetranslator` automatisch im Sortierungsablauf betrachtet werden.

```
\DeclareSortingScheme{sample}{ \sort{
\name{author} \name{editor} \name{translator} }
\sort{\field{title} }
\sort{ \field{year} } }
```

Im nächsten Beispiel definieren wir das selbe Schema in einem weiter ausgearbeiteten Weg, unter Berücksichtigung solcher Fachbereiche, wie `presort`, `sortkey`, `sortname`, etc. Da das `sortkey`-Datenfeld den Hauptgattungsschlüssel angibt, muss es alle anderen Elemente, abgesehen von `presort`, ausschalten. Dies wird von der `final`-Option angegeben. Wenn der `sortkey` verfügbar ist, dann wird der Prozess an diesem Punkt gestoppt. Wenn nicht, dann geht das Sortierungsprogramm mit der nächsten `\sort`-Anordnung weiter. Die Einstellung entspricht der Hauptprogramm-Definition vom `nty`-Schema.

```
\DeclareSortingScheme{nty}{ \sort{
\field{presort} } \sort[final]{ \field{sortkey} }
```

```

\sort{\name{sortname} \name{author} \name{editor} \name{translator}
\field{sorttitle}
\field{title} } \sort{ \field{sorttitle} \field{title} }
\sort{ \field{sortyear} \field{year} } }

```

`\DeclareSortExclusion{⟨entrytype, ...⟩}{⟨field, ...⟩}`

Biber only

Spezifiziert Bereiche, die vom Sortieren pro Tastenschlag ausgeschlossen sind. Das Argument `⟨entrytype⟩` und das Argument `⟨field⟩` können eine Komma-getrennte Liste von Werten sein. Ein leeres `⟨field⟩`-Argument wird alle Ausschlüsse für diesen `⟨entrytype⟩` ausgleichen. Dieser Befehl kann nur für die Präambel genutzt werden.

`\DeclarePresort[⟨entrytype, ...⟩]{⟨string⟩}`

Biber only

Spezifizierungen sind Abläufe, die dazu verwendet werden, den `presort`-Bereich der Einträge automatisch, ohne den `presort`-Bereich zu erweitern. Das `presort` kann allgemein oder pro Tastenschlag definiert sein. Wenn das optionale Argument `⟨entrytype⟩` gegeben ist, dann gilt der `⟨string⟩` für den jeweiligen Zugangstyp. Wenn nicht, dann dient es als der allgemeine Vorzugswert. Das Spezifizieren von `⟨entrytype⟩` in Verbindung mit einem leeren `⟨string⟩` wird die typspezifische Einstellung löschen. Das Argument `⟨entrytype⟩` kann eine Komma-getrennte Liste von Werten sein. Dieser Befehl kann nur in der Präambel benutzt werden.

#### 4.5.2. Spezielle Felder

Manche der automatisch generierten Felder aus § 4.2.4.2 können angepasst werden. Beachten Sie, dass dies Biber benötigt.

`\DeclareLabelname[⟨entrytype, ...⟩]{⟨specification⟩}`

Biber only

Definiert die Felder, die betrachtet werden, wenn das `labelname` erstellt wird (siehe § 4.2.4.2). Das `⟨specification⟩` Argument ist eine durch Komma-getrennte Liste von Feldern. Die Felder werden in dieser Reihenfolge überprüft und das erste Felde, das verfügbar ist, wird als `labelname` benutzt. Dies ist die Standarddefinition:

```

\DeclareLabelname{shortauthor,author,shorteditor,editor,translator}

```

Das `labelname`-Feld kann global oder für einen Typ angepasst werden. Ist das optionale `⟨entrytype⟩` Argument gegeben, dann wird die Spezifikation auf den jeweiligen Eintragstyp angewendet. Falls nicht, dann wird sie global angewendet. Das `⟨entrytype⟩`-Argument kann eine durch Komma-getrennte Liste von Werten sein. Dieser Befehl kann nur in der Präambel genutzt werden.

`\DeclareLabelyear` [*entrytype*, ... ] {*specification*}

Biber only

Definiert die Datumskomponenten, die betrachtet werden, wenn das `labelyear`-Feld erstellt wird (siehe § 4.2.4.2). Das *specification* Argument ist eine durch Komma-getrennte Liste von Datumskomponenten. Die Elemente werden in dieser Reihenfolge überprüft und das erste Element, das verfügbar ist, wird als `labelyear` genutzt. Beachten Sie, dass diese Elemente Datumskomponenten und keine Datumsfelder sind. Dies ist die Standarddefinition:

```
\DeclareLabelyear{year,eventyear,origyear,urlyear}
```

Das `labelyear`-Feld kann global oder für einen Typ angepasst werden. Ist das optionale *entrytype* Argument gegeben, dann wird die Spezifikation auf den entsprechenden Eintragstypen angewendet. Falls nicht, dann wird Sie globale angewendet. Das *entrytype* kann eine durch Komma-getrennte Liste von Werten sein. Dieser Befehl kann nur in der Prämbel genutzt werden. Siehen Sie auch § 4.2.4.3.

### 4.5.3. Datenvererbung (crossref)

Biber unterstützt einen hochgradig anpassbarer Mechanismus für Querverweise mit flexiblen Vererbungsregeln. Diese Abschnitte befasst sich mit der Konfigurationsoberfläche. Siehe appendix A für die Standardkonfiguration. Beachten Sie, dass eine benutzerdefinierte Datenvererbung Biber erfordert. Es funktioniert mit keiner anderen Erweiterung. Ein Hinweis zur Terminologie: *child* oder *target* ist der Eintrag mit dem `crossref`-Feld, *parent* oder *source* ist der Eintrag, auf den das `crossref`-Feld zeigt. Der untergeordnete Eintrag erbt Daten von dem übergeordneten Eintrag.

`\DefaultInheritance` [*exceptions*] {*options*}

Biber only

Konfiguriert das standardmäßige Vererbungsverhalten. Dieser Befehl darf nur in der Prämbel benutzt werden. Das Standardverhalten kann durch die folgenden *options* angepasst werden:

```
all=true, false default: true
```

Sollen alle oder keine Felder vererbt werden. `all=true` bedeutet, dass der untergeordnete Eintrag alle Werte von dem übergeordneten Eintrag erbt, solange keine genauere Vererbungsregel durch `\DeclareDataInheritance` definiert wurde. Ist eine Vererbungsregel für ein Feld definiert, so wird die Datenvererbung durch diese Regel gesteuert. `all=false` bedeutet, dass standardmäßig keine Daten vererbt werden. Jedes Feld, das vererbt werden soll, benötigt eine genaue Vererbungsregel, die mit `\DeclareDataInheritance` festgelegt werden muss. Der Standardwert für ein Package ist `all=true`.

`override=true, false` default: false

Ob die *target*-Felder von den *source*-Feldern überschrieben werden sollen, wenn beide definiert sind. Dies gilt sowohl für die automatische Vererbung, als auch für explizite Vererbungsregeln. Der Standardwert des Paketes ist `override=false`, sodass z. B. schon existierende Felder des untergeordneten Eintrags nicht überschrieben werden.

Die zusätzlichen *exceptions* sind eine Liste von `\except` Richtlinien. Leerzeichen, Tabulatoren, und Zeilenumbrüche können beliebig benutzt werden, um die *exceptions* zu ordnen. Leere Zeilen sind nicht erlaubt.

`\except{<source>}{<target>}{<options>}`

Legt die *options* für eine bestimmte *source*-*target*-Kombination fest. Die *source* und *target* Argumente legen den über- und untergeordneten Eintragstyp fest. Der Asterisk entspricht allen Typen und kann in beiden Argumenten verwendet werden.

`\DeclareDataInheritance{<source, ...>}{<target, ...>}{<rules>}`

Biber only

Deklariert Vererbungsregeln. Die *source* und *target* Argumente legen den über- und untergeordneten Eintragstyp fest. Jedes Argument kann ein einzelner Eintragstyp, eine durch Komma-getrennte Liste von Typen oder ein Asterisk sein. Ein Asterisk entspricht allen Typen. Die *rules* sind eine unbegrenzte Liste von `\inherit` und/oder `\noinherit` Richtlinien. Leerzeichen, Tabulatoren und Zeilenumbrüche können beliebig benutzt werden, um die *rules* visuell darzustellen. Leere Zeilen sind nicht erlaubt. Dieser Befehl darf nur in der Prämbel verwendet werden.

`\inherit[<option>]{<source>}{<target>}`

Definiert eine Vererbungsregel, durch die das *source*-Feld auf das *target*-Feld abgebildet wird. Das *option*-Feld stellt die oben erklärte `override`-Option zur Verfügung. Wird die Regel lokal definiert, so hat Sie Vorrang vor jeder globalen Vererbungsregel, die mit `\DefaultInheritance` festgelegt wurde.

`\noinherit{<source>}`

Verhindert bedingungslos die Vererbung des *source*-Feldes.

`\ResetDataInheritance` Löscht alle. Dieser Befehl darf nur in der Prämbel benutzt werden.

Biber only

Dazu einige praktische Beispiele:

```
\DefaultInheritance{all=true,override=false}
```

Dieses Beispiel zeigt, wie man das Standardverhalten der Vererbung festlegt. Die obigen Einstellungen sind die Standardwerte für ein Paket.

```
\DefaultInheritance[\except{*}{online}{all=false}]
{all=true,override=false}
```

Dieses Beispiel ist dem obigen ähnlich, fügt jedoch eine Ausnahme hinzu: Einträge des Typs @online erben standardmäßig keine Daten von übergeordneten Einträgen.

```
\DeclareDataInheritance{collection}{incollection}{
\inherit{title}{booktitle} \inherit{subtitle}{booksubtitle}
\inherit{titleaddon}{booktitleaddon} }
```

Bisher haben wir uns angeschaut, wie man die Standardeinstellungen setzt. Z. B. bedeutet `all=true`, dass das `publisher`-Feld von dem `source`-Eintrag in das `publisher`-Feld des `target`-Eintrags kopiert wird. In manchen Fällen werden aber auch asymmetrische Abbildungen benötigt. Sie werden mit `\DeclareDataInheritance` definiert. Das obige Beispiel setzt 3 typische Regeln für @incollection Einträge, die auf eine @collection verweisen. Wir weisen den `source`-Feldern (`title`, ...) die entsprechenden `target`-Felder (`booktitle`, ...) zu.

```
\DeclareDataInheritance{mvbook,book}{inbook,bookinbook}{
\inherit{author}{author} \inherit{author}{bookauthor} }
```

Diese Regel ist ein Beispiel für eine Abbildung von einem auf mehrere andere Einträge: Das `author`-Feld als Quelle und für das `author`- und das `bookauthor`-Feld des Zieles, um kompakte `inbook`/`bookinbook`-Einträge zu ermöglichen. Die Quelle kann entweder ein @mvbook- oder ein @book-Eintrag sein. Das Ziel kann entweder ein @inbook oder ein @bookinbook-Eintrag sein.

```
\DeclareDataInheritance{*}{inbook,incollection}{
\noinherit{introduction} }
```

Diese Regel verhindert die Vererbung des `introduction`-Feldes. Es gilt für alle `target`-Einträge des Typs @inbook oder @incollection, unabhängig von den Typen der `source`-Einträge.

```
\noinherit{abstract} }
```

Diese Regel, die für alle Einträge unabhängig von deren Typ gilt, verhindert die Vererbung des `abstract`-Feldes.

```
\ResetDataInheritance
```

Dieses Beispiel zeigt, wie man BibTeX's Querverweismechanismus emuliert. Es aktiviert die Vererbung, deaktiviert das Überschreiben und löscht alle anderen Vererbungsregeln und Abbildungen.

## 4.6. Hilfsbefehle

Die Einrichtungen in diesem Bereich sind für die Analyse und Speicherung bibliographischen Daten vorgesehen und nicht für das Formatieren und Drucken.

### 4.6.1. Datenbefehle

Die Befehle in diesem Bereich gewährleisten den schwachen Zugriff auf die unformatierten bibliographischen Daten. Sie sind nicht für das Setzen der Schriften bestimmt, sondern beispielsweise für das Speichern der Daten in einem temporären Makro, welches später in einem Vergleich verwendet werden kann.

`\thefield{⟨field⟩}`

Erweitert um das unformatierte *⟨field⟩*. Wenn *⟨field⟩* nicht definiert ist, dann erweitert dieser Befehl um einen leeren String.

`\strfield{⟨field⟩}`

Ähnlich zu `\thefield`, außer dass das Feld automatisch zensiert wird, sodass dessen Wert die Bildung eines Kontroll-Sequenz-Namens sichert.

`\thelist{⟨literal list⟩}`

Erweitert um die unformatierte *⟨literal list⟩*. Wenn die Liste undefiniert ist, dann erweitert dieser Befehl um einen leeren String. Beachten Sie, dass dieser Befehl die *⟨literal list⟩* in einem internen Format, welches von diesem Paket genutzt wird, lagert. Dieses Format ist nicht für das Drucken geeignet.

`\thename{⟨name list⟩}`

Erweitert um die unformatierte *⟨name list⟩*. Wenn die Liste undefiniert ist, dann erweitert dieser Befehl um einen leeren String. Beachten Sie, dass dieser Befehl die *⟨name list⟩* in einem internen Format, welches von diesem Paket genutzt wird, lagert. Dieses Format ist nicht für das Drucken geeignet.

`\savefield{⟨field⟩}{⟨macro⟩}`

`\savefield*{⟨field⟩}{⟨macro⟩}`

Kopiert das unformatierte *⟨field⟩* in ein *⟨macro⟩*. Die reguläre Variante dieses Befehls definiert das *⟨macro⟩* global, die Variante mit dem Sternchen hingegen lokal.

`\savelist{⟨literal list⟩}{⟨macro⟩}`

`\savelist*{⟨literal list⟩}{⟨macro⟩}`

Kopiert die unformatierte *⟨literal list⟩* in ein *⟨macro⟩*. Die reguläre Variante dieses Befehls definiert das *⟨macro⟩* global, die Variante mit dem Sternchen hingegen lokal.

```
\savename{<name list>}{<macro>}
\savename*{<name list>}{<macro>}
```

Kopiert das unformatierte *<name list>* in ein *<macro>*. Die reguläre Variante dieses Befehls definiert das *<macro>* global, die Variante mit dem Sternchen hingegen lokal.

```
\savefieldcs{<field>}{<cname>}
\savefieldcs*{<field>}{<cname>}
```

Ähnlich wie `\savefield`, nimmt jedoch den Kontroll-Sequenz-Namen *<cname>* (ohne führenden Backslash) und nicht den Makronamen als Argument.

```
\savelistcs{<literal list>}{<cname>}
\savelistcs*{<literal list>}{<cname>}
```

Ähnlich wie `\savelist`, nimmt jedoch den Kontroll-Sequenz-Namen *<cname>* (ohne führenden Backslash) und nicht den Makronamen als Argument.

```
\savenamecs{<name list>}{<cname>}
\savenamecs*{<name list>}{<cname>}
```

Ähnlich wie `\savename`, nimmt jedoch den Kontroll-Sequenz-Namen *<cname>* (ohne führenden Backslash) und nicht den Makronamen als Argument.

```
\restorefield{<field>}{<macro>}
```

Stellt ein *<field>* von einem *<macro>*, welches mit `\savefield` definiert wurde, wieder her. Das Feld wird in lokalen Bereich wiederhergestellt.

```
\restorelist{<literal list>}{<macro>}
```

Stellt ein *<literal list>* von einem *<macro>*, welches mit `\savelist` definiert wurde, wieder her. Das Feld wird in lokalen Bereich wiederhergestellt.

```
\restorename{<name list>}{<macro>}
```

Stellt ein *<name list>* von einem *<macro>*, welches mit `\savename` definiert wurde, wieder her. Das Feld wird in lokalen Bereich wiederhergestellt.

```
\clearfield{<field>}
```

Löscht *<field>* innerhalb eines lokalen Bereichs. Ein Feld, das so gelöscht wurde, wird von nachfolgenden Datenbefehlen als undefiniert behandelt.

```
\clearlist{<literal list>}
```

Löscht *<literal list>* innerhalb eines lokalen Bereichs. Eine Liste, die so gelöscht wurde, wird von nachfolgenden Datenbefehlen als undefiniert behandelt.

`\clearname{⟨name list⟩}`

Löscht *⟨name list⟩* innerhalb eines lokalen Bereichs. Eine Liste, die so gelöscht wurde, wird von nachfolgenden Datenbefehlen als undefiniert behandelt.

#### 4.6.2. Eigenständige Tests

Die Befehle in diesem Abschnitt sind verschiedene Arten von eigenständigen Tests, die in Bibliographien und Zitierstilen benutzt werden.

`\iffieldundef{⟨field⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls *⟨field⟩* nicht definiert ist, ansonsten um *⟨false⟩*.

`\iflistundef{⟨literal list⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls *⟨literal list⟩* nicht definiert ist, ansonsten um *⟨false⟩*.

`\ifnameundef{⟨name list⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls *⟨name list⟩* nicht definiert ist, ansonsten um *⟨false⟩*.

`\iffieldsequal{⟨field 1⟩}{⟨field 2⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls die Werte von *⟨field 1⟩* und *⟨field 2⟩* gleich sind, ansonsten um *⟨false⟩*.

`\iflistsequal{⟨literal list 1⟩}{⟨literal list 2⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls die Werte von *⟨literal list 1⟩* und *⟨literal list 2⟩* gleich sind, ansonsten um *⟨false⟩*.

`\ifnamesequal{⟨name list 1⟩}{⟨name list 2⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls die Werte von *⟨name list 1⟩* und *⟨name list 2⟩* gleich sind, ansonsten um *⟨false⟩*.

`\iffieldequals{⟨field⟩}{⟨macro⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls der Wert von *⟨field⟩* und die Definition von *⟨macro⟩* gleich sind, ansonsten um *⟨false⟩*.

`\iflistequals{⟨literal list⟩}{⟨macro⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls der Wert von *⟨literal list⟩* und die Definition von *⟨macro⟩* gleich sind, ansonsten um *⟨false⟩*.

`\ifnameequals`{*<name list>*}{*<macro>*}{*<true>*}{*<false>*}

Erweitert um *<true>*, falls der Wert von *<name list>* und die Definition von *<macro>* gleich sind, ansonsten um *<false>*.

`\iffieldequalcs`{*<field>*}{*<csname>*}{*<true>*}{*<false>*}

Ähnlich wie `\iffieldequals`, nimmt jedoch den Kontroll-Sequenz-Namen *<csname>* (ohne führenden Backslash) anstatt den *<macro>*-Namen als Argument.

`\iflistequalcs`{*<literal list>*}{*<csname>*}{*<true>*}{*<false>*}

Ähnlich wie `\iflistequals`, nimmt jedoch den Kontroll-Sequenz-Namen *<csname>* (ohne führenden Backslash) anstatt dem *<macro>* Namen als Argument.

`\ifnameequalcs`{*<name list>*}{*<csname>*}{*<true>*}{*<false>*}

Ähnlich wie `\ifnameequals`, nimmt jedoch den Kontroll-Sequenz-Namen *<csname>* (ohne führenden Backslash) anstatt den *<macro>*-Namen als Argument.

`\iffieldequalstr`{*<field>*}{*<string>*}{*<true>*}{*<false>*}

Führt *<true>* aus, falls der Wert von *<field>* gleich *<string>* ist, ansonsten *<false>*. Dies ist ein starker Befehl.

`\iffieldxref`{*<field>*}{*<true>*}{*<false>*}

Falls das `crossref/xref`-Feld eines Eintrags definiert ist, überprüft dieser Befehl, ob *<field>* mit einem durch einen Querverweis übergeordneten Eintrag verbunden ist. Er führt *<true>* aus, falls *<field>* des untergeordneten Eintrags gleich dem entsprechenden *<field>* des übergeordneten Eintrags ist. Ansonsten wird *<false>* ausgeführt. Ist das `crossref/xref`-Feld undefiniert, so wird *<false>* ausgeführt. Dies ist ein robuster Befehl. Für weitere Informationen über Querverweise lesen Sie die Beschreibungen der `crossref` und `xref` Felder in § 2.2.3 sowie § 2.4.1.

`\iflistxref`{*<literal list>*}{*<true>*}{*<false>*}

Ähnlich wie `\iffieldxref`, überprüft jedoch, ob *<literal list>* mit einem durch einen Querverweis übergeordneten Eintrag verbunden ist. Für weitere Informationen über Querverweise lesen Sie die Beschreibungen der `crossref` und `xref` Felder in § 2.2.3 sowie § 2.4.1.

`\ifnamexref`{*<name list>*}{*<true>*}{*<false>*}

Ähnlich wie `\iffieldxref`, überprüft jedoch, ob *<name list>* mit einem durch einen Querverweis übergeordneten Eintrag verbunden ist. Für weitere Informationen über Querverweise lesen Sie die Beschreibungen der `crossref` und `xref` Felder in § 2.2.3 sowie § 2.4.1.

`\ifcurrentfield{⟨field⟩}{⟨true⟩}{⟨false⟩}`

Führt *⟨true⟩* aus, falls *⟨field⟩* das aktuelle Feld ist. Ansonsten wird *⟨false⟩* ausgeführt. Dies ist ein robuster Befehl. Er ist für die Verwendung in Feld-Formatierungs-Richtlinien gedacht und führt *⟨false⟩* aus, wenn er anders verwendet wird.

`\ifcurrentlist{⟨literal list⟩}{⟨true⟩}{⟨false⟩}`

Führt *⟨true⟩* aus, falls *⟨literal list⟩* die aktuelle Liste ist. Ansonsten wird *⟨false⟩* ausgeführt. Dies ist ein robuster Befehl. Er ist für die Verwendung in Listen-Formatierungs-Richtlinien gedacht und führt *⟨false⟩* aus, wenn er anders verwendet wird.

`\ifcurrentname{⟨name list⟩}{⟨true⟩}{⟨false⟩}`

Führt *⟨true⟩* aus, falls *⟨name list⟩* die aktuelle Liste ist. Ansonsten wird *⟨false⟩* ausgeführt. Dies ist ein robuster Befehl. Er ist für die Verwendung in Listen-Formatierungs-Richtlinien gedacht und führt *⟨false⟩* aus, wenn er anders verwendet wird.

`\ifuseprefix{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls die `useprefix`-Option aktiviert ist (entweder global oder für den aktuellen Eintrag), ansonsten wird *⟨false⟩* ausgeführt. Siehe § 3.1.3 für Details über diese Option.

`\ifuseauthor{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls die `useauthor`-Option aktiviert ist (entweder global oder für den aktuellen Eintrag), ansonsten wird *⟨false⟩* ausgeführt. Siehe § 3.1.3 für Details über diese Option.

`\ifuseeditor{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls die `useeditor`-Option aktiviert ist (entweder global oder für den aktuellen Eintrag), ansonsten wird *⟨false⟩* ausgeführt. Siehe § 3.1.3 für Details über diese Option.

`\ifusetranslator{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls die `usetranslator`-Option aktiviert ist (entweder global oder für den aktuellen Eintrag), ansonsten wird *⟨false⟩* ausgeführt. Siehe § 3.1.3 für Details über diese Option.

`\ifsingletitle{⟨true⟩}{⟨false⟩}`

Erweitert um *⟨true⟩*, falls in der Bibliographie nur ein Werk von dem Ersteller/Bearbeiter existiert, ansonsten wird um *⟨false⟩* erweitert. Beachten Sie, dass dieses Feature explizit mit der Package-Option `singletitle` aktiviert werden muss.

`\ifandothers{⟨list⟩}{⟨true⟩}{⟨false⟩}`

Erweitert um `⟨true⟩`, falls die `⟨list⟩` definiert ist und in der `bib` Datei mit dem Schlüsselwort `,and others'` abgekürzt wurde, ansonsten wird um `⟨false⟩` erweitert. Die `⟨list⟩` kann eine Buchstabenliste oder Namenliste sein.

`\ifmorenames{⟨true⟩}{⟨false⟩}`

Erweitert um `⟨true⟩`, falls die aktuelle Namensliste abgeschnitten wurde oder noch abgeschnitten wird. Ansonsten wird um `⟨false⟩` erweitert. Dieser Befehl ist für die Verwendung in Formatierungsrichtlinien für Namenslisten bestimmt. Wird der Befehl in einem anderen Zusammenhang verwendet, erweitert er immer um `⟨false⟩`. Dieser Befehl führt das Äquivalent eines `\ifandothers`-Tests für die aktuelle Liste aus. Fällt dieser Test negativ aus, so überprüft er außerdem, ob der `listtotal`-Zähler größer als `liststop` ist. Dieser Befehl kann in einer Formatierungsrichtlinie genutzt werden, um zu entscheiden, ob eine Notiz wie „and others“ oder „et al.“ am Ende der Liste gedruckt wird. Beachten Sie, dass Sie überprüfen müssen, ob Sie in der Mitte oder am Ende der Liste sind, z. B. ob `listcount` kleiner oder gleich `liststop` ist. Siehe § 4.4.1 für Details.

`\ifmoreitems{⟨true⟩}{⟨false⟩}`

Dieser Befehl ist ähnlich wie `\ifmorenames`, überprüft jedoch die aktuelle Buchstabenliste. Er ist für die Verwendung in Formatierungsrichtlinien für Buchstabenlisten. Der Befehl wird immer um `⟨false⟩` erweitert, wenn er anders verwendet wird.

`\iffirstinits{⟨true⟩}{⟨false⟩}`

Erweitert um `⟨true⟩` oder `⟨false⟩`, abhängig von dem Status der `firstinits` Package Option (Siehe § 3.1.2.3). Dieser Befehl ist für die Verwendung in Formatierungsrichtlinien für Namenslisten bestimmt.

`\ifkeyword{⟨keyword⟩}{⟨true⟩}{⟨false⟩}`

Führt `⟨true⟩` aus, falls `⟨keyword⟩` in dem `keywords`-Feld des Eintrags, der gerade bearbeitet wird, gefunden wird. Ansonsten wird `⟨false⟩` ausgeführt.

`\ifentrykeyword{⟨entrykey⟩}{⟨keyword⟩}{⟨true⟩}{⟨false⟩}`

Eine Variante von `\ifkeyword`, welche einen Schlüsseleintrag als erstes Argument nimmt. Dies ist nützlich, wenn man einen anderen Eintrag als den Eintrag, der gerade bearbeitet wird, testen will.

`\ifcategory{⟨category⟩}{⟨true⟩}{⟨false⟩}`

Führt `⟨true⟩` aus, falls der Eintrag, der gerade bearbeitet wird, einer `⟨category⟩` mit `\addtcategory` zugewiesen wurde. Ansonsten wird `⟨false⟩` ausgeführt.

`\ifentrycategory{⟨entrykey⟩}{⟨category⟩}{⟨true⟩}{⟨false⟩}`

Eine Variante von `\ifcategory`, welche einen Schlüsseleintrag als erstes Argument nimmt. Dies ist nützlich, wenn man einen anderen Eintrag als den Eintrag, der gerade bearbeitet wird, testen will.

`\ifciteseen{⟨true⟩}{⟨false⟩}`

Führt `⟨true⟩` aus, falls der Eintrag, der gerade bearbeitet wird, bereits zitiert wurde. Ansonsten wird `⟨false⟩` ausgeführt. Dies ist ein robuster Befehl und für die Verwendung in Zitierstilen bestimmt. Falls es irgendwelche `refsection`-Umgebungen in diesem Dokument gibt, dann ist die Zitatverfolgung für diese Umgebungen lokal. Beachte, dass die Zitatverfolgung mit der Package Option `citetracker` aktiviert sein muss. Das Verhalten dieses Tests hängt von dem Modus, in dem die Zitatverfolgung arbeitet, ab. Siehe § 3.1.2.3 für Details. Ist die Zitatverfolgung ausgeschaltet, dann ist das Ergebnis des Tests `⟨false⟩`. Beachte auch, dass `\citetrackertrue` und `\citetrackerfalse` sich in § 4.6.4 vertauschen.

`\ifentryseen{⟨entrykey⟩}{⟨true⟩}{⟨false⟩}`

Eine Variante von `\ifciteseen`, welche einen Schlüsseleintrag als erstes Argument nimmt. Da `⟨entrykey⟩` vor der Durchführung des Tests erweitert ist, ist es möglich, Schlüsseleinträge in einem Feld wie `xref` zu testen:

```
\ifentryseen{\thefield{xref}}{true}{false}
```

Abgesehen von dem zusätzlichem Argument verhält sich `\ifentryseen` wie `\ifciteseen`.

`\ifciteibid{⟨true⟩}{⟨false⟩}`

Erweitert um `⟨true⟩`, falls der Eintrag, der gerade bearbeitet wird, der selbe ist, wie der letzte Eintrag. Ansonsten wird `⟨false⟩` ausgeführt. Dieser Befehl ist für die Verwendung in Zitierstilen bestimmt. Gibt es in dem Dokument irgendwelche `refsection`-Umgebungen, dann erfolgt die Verfolgung in diesen Umgebungen lokal. Beachten Sie, dass die ‚ibidem‘-Verfolgung mit der Package-Option `ibidtracker` aktiviert sein muss. Das Verhalten dieses Tests hängt von dem Modus ab, in dem der Verfolger arbeitet. Siehe § 3.1.2.3 für Details. Ist der Verfolger deaktiviert, liefert der Test immer `⟨false⟩`. Siehe außerdem `\citetrackertrue` und `\citetrackerfalse` in § 4.6.4.

`\ifciteidem{⟨true⟩}{⟨false⟩}`

Erweitert um `⟨true⟩`, falls der Primärname (z. B. Author oder Editor) des Eintrags, der gerade bearbeitet wird, der gleiche ist wie der des letzten Eintrags. Ansonsten

wird um  $\langle false \rangle$  erweitert. Dieser Befehl ist für die Verwendung in Zitierstilen bestimmt. Gibt es irgendwelche `refsection`-Umgebungen in dem Dokument, dann ist der Verfolger für diese Umgebungen lokal. Beachte, dass der ‚idem‘-Verfolger mit der Package-Option `idemtracker` aktiviert sein muss. Das Verhalten dieses Tests hängt von dem Modus ab, in dem der Verfolger arbeitet. Siehe § 3.1.2.3 für Details. Ist der Verfolger deaktiviert, liefert der Test immer  $\langle false \rangle$ . Siehe außerdem `\citetrackertrue` und `\citetrackerfalse` in § 4.6.4.

`\ifopcit` $\langle true \rangle$  $\langle false \rangle$

Dieser Befehl ist ähnlich wie `\ifciteibid`, außer dass es um  $\langle true \rangle$  erweitert, wenn der Eintrag, der gerade bearbeitet wird, der gleiche ist wie der letzte Eintrag *von diesem Ersteller oder Bearbeiter*. Beachte, dass der ‚opcit‘-Verfolger mit der Package-Option `opcitracker` aktiviert sein muss. Das Verhalten dieses Tests hängt von dem Modus ab, in dem der Verfolger arbeitet. Siehe § 3.1.2.3 für Details. Ist der Verfolger deaktiviert, liefert der Test immer  $\langle false \rangle$ . Beachte außerdem, dass sich `\citetrackertrue` und `\citetrackerfalse` in § 4.6.4 vertauschen.

`\ifloccit` $\langle true \rangle$  $\langle false \rangle$

Dieser Befehl ist ähnlich wie `\ifopcit`, außer dass er zusätzlich die  $\langle postnote \rangle$ -Argumente vergleicht und nur um  $\langle true \rangle$  erweitert, wenn Sie gleich und numerisch sind (im Sinne von `\ifnumerals` aus § 4.6.2). Beispielsweise liefert `\ifloccit true`, falls das Zitat sich auf die selbe Seite bezieht, wie das letzte Zitat. Beachte, dass der ‚loccit‘-Verfolger mit der Package-Option `loccitracker` aktiviert sein muss. Das Verhalten dieses Tests hängt von dem Modus ab, in dem der Verfolger arbeitet. Siehe § 3.1.2.3 für Details. Ist der Verfolger deaktiviert, liefert der Test immer  $\langle false \rangle$ . Beachte außerdem, dass sich `\citetrackertrue` und `\citetrackerfalse` in § 4.6.4 vertauschen.

`\iffirstonpage` $\langle true \rangle$  $\langle false \rangle$

Das Verhalten dieses Befehls reagiert auf die Package-Option `pagetracker`. Ist diese Option auf `page` gesetzt, dann erweitert der Befehl um  $\langle true \rangle$ , falls das aktuelle Element das erste auf der Seite ist, ansonsten um  $\langle false \rangle$ . Ist diese Option auf `spread` gesetzt, dann erweitert der Befehl um  $\langle true \rangle$ , falls das aktuelle Element das erste auf der Doppelseite ist, ansonsten um  $\langle false \rangle$ . Ist der Seitenverfolger ausgeschaltet, dann liefert der Test  $\langle false \rangle$ . Abhängig von dem Zusammenhang kann das Element ein Zitat, ein Eintrag in einer Bibliographie oder eine Liste von Shorthands. Beachte, dass dieser Test zwischen Fließtext und Fußnoten unterscheidet. Wird er ist 0, falls der Nachname des Ersteller oder Bearbeiter einzigartig ist, 1, falls das Hinzufügen von anderen Teilen des Namens (Vorname, Prefix, Suffix) als Initialen ihn einzigartig machen, und 2, falls der volle Name benötigt wird, um ihn einzigartig zu machen. Diese Information wird von dem Ersteller-Jahr- und dem Ersteller-Titel-Zitierschema

benötigt, welches, wenn verschiedene Ersteller mit dem selben Nachnamen zitiert werden, zusätzliche Teile des Namens hinzufügt. Gibt es z. B. in der Liste der Referenzen einen ‚John Doe‘ und einen ‚Edward Doe‘, dann wird der Counter auf 1 gesetzt. Gibt es einen ‚John Doe‘ und eine ‚Jane Doe‘, dann wird der Counter auf 2 gesetzt. Ist die Option auf `init` gesetzt, dann wird der Counter auf 1 begrenzt. Dies ist für Zitierstile nützlich, die Initialen verwenden, um Namen eindeutig zu machen, aber niemals den vollen Namen in Zitaten drucken. Wenn das Hinzufügen der Initialen nicht reicht, um den Namen einzigartig zu machen, dann wird `uniquename` für diesen Namen auf 0 gesetzt. Dieses Feature muss mit der Package-Option `uniquename` aktiviert sein. Beachte, dass der `uniquename` Zähler ‚innerhalb‘ `\printnames` und nur für die `labelname` Liste gesetzt ist. Dessen Wert ist 0 in jedem anderen Zusammenhang, z. B. muss er in der Namen-Formatierungsrichtlinie, die mit der `labelname`-Liste arbeitet, ausgewertet werden.

#### `uniquelist`

Biber only

Dieser Zähler bezieht sich auf die `labelname`-Liste und ist auf eine per-Eintrag-Basis gesetzt. Dessen Wert zeigt die Anzahl an Namen, die benötigt wird, um die `labelname`-Liste einzigartig zu machen, falls die automatische `maxnames`/`minnames`-Kürzung zu einer mehrdeutigen Liste von Namen führen würde. Gibt es z. B. eine Arbeit von ‚Doe/Smith/Johnson‘ und eine andere Arbeit von ‚Doe/Edwards/Williams‘, dann könnte das Setzen von `maxnames=1` beim Zitieren in beiden Fällen zu ‚Doe et al.‘ führen (abhängig von dem Stil). In diesem Fall würde `uniquelist` für beide Einträge auf 2 gesetzt sein, da mindestens die ersten beiden Namen benötigt werden, um die die Einträge der `labelname` Liste einzigartig zu machen. Beachte, dass der `uniquelist` Zähler ‚innerhalb‘ `\printnames` und nur für die `labelname`-Liste gesetzt ist. Dessen Wert ist in jedem anderen Zusammenhang 0. Ist der `uniquelist` Wert verfügbar und wird die `labelname`-Liste bearbeitet, dann wird er von `\printnames` genutzt, z. B. überschreibt er automatisch `maxnames`. Dieses Feature muss mit der Package-Option `uniquelist` aktiviert sein.

`parenlevel` Die aktuelle Schachtelungsebene von runden und/oder eckigen Klammern. Diese Information ist nur dann verfügbar, wenn die `parenttracker`-Option aus § 3.1.2.3 aktiviert ist.

#### 4.6.3. Tests mit `\ifbool` und `\ifthenelse`

Diese Tests, die in § 4.6.2 vorgestellt wurden, können auch mit dem `\ifbool`-Befehl, der von dem `etoolbox`-Paket zur Verfügung gestellt wird und dem `\ifthenelse` Befehl, der von dem `ifthen`-Paket zur Verfügung gestellt wird, genutzt werden.

Der Syntax dieser Tests ist in diesem Fall etwas anders: Die `<true>` und `<false>` Argumente werden von den Tests selbst ausgelassen und stattdessen an den `\ifbool` oder `\ifthenelse` Befehl übergeben. Beachte, dass die Benutzung dieser Befehle Ar-

beitsaufwand verursacht. Falls du keine booleschen Operatoren benötigst, dann ist es effizienter, die selbstständigen Tests aus § 4.6.2 zu benutzen.

`\ifboolexpr{<expression>}{<true>}{<false>}`

etoolbox Befehl, der komplexe Tests mit booleschen Operatoren und Gruppen erlaubt:

```
\ifboolexpr{ (test {\ifnameundef{editor}}
and not test {\iflistundef{location}}) or test {\iffieldundef{year}} } {...}
{...}
```

`\ifthenelse{<tests>}{<true>}{<false>}`

ifthen Befehl, der komplexe Tests mit booleschen Operatoren und Gruppen erlaubt:

```
\ifthenelse{ \ (\ifnameundef{editor} \and
\not \iflistundef{location} \) \or \iffieldundef{year} } {...} {...}
```

Die zusätzlichen Tests, die von biblatex zur Verfügung gestellt werden, sind nur dann verfügbar, wenn `\ifboolexpr` oder `\ifthenelse` in Zitierbefehlen und in der Bibliographie benutzt werden.

#### 4.6.4. Verschiedene Befehle

Dieser Abschnitt stellt verschiedene Befehle und kleine Helfer zur Verfügung, die in der Bibliographie und in Zitierstilen benutzt werden.

`\newbibmacro{<name>}[<arguments>][<optional>]{<definition>}`  
`\newbibmacro*{<name>}[<arguments>][<optional>]{<definition>}`

Definiert ein Makro, welches später via `\usebibmacro` ausgeführt wird. Der Syntax dieses Befehls ist `\newcommand` sehr ähnlich, außer dass `<name>`-Zeichen wie Zahlen und Satzzeichen enthalten sein können und man nicht mit einem Backslash starten darf. Das optionale Argument `<arguments>` ist ein Integer, der die Anzahl an Argumenten in dem Makro enthält. Existiert `<optional>`, so gibt es den Standardwert des ersten Arguments des Makro an, welches automatisch zu einem optionalen Argument wird. Im Gegensatz zu `\newcommand` bringt `\newbibmacro` eine Warnmeldung, falls ein Makro bereits definiert ist und fällt automatisch in `\renewbibmacro` zurück. Genauso wie `\newcommand` benutzt die reguläre Variante dieses Befehls das `\long` in der Definition, solange die Variante mit dem Asterisk dies nicht tut. Wurde ein Makro lang deklariert, so kann es Argumente nehmen, die `\par` Marken enthalten. `\newbibmacro` and `\renewbibmacro` werden zweckmäßig zur Verfügung gestellt. Stilersteller können stattdessen auch `\newcommand` oder `\def` benutzen. Beachten

Sie trotzdem, dass die meisten gemeinsamen Definitionen, die sich in `biblatex.def` befinden, mit `\newbibmacro` definiert sind, damit sie entsprechend benutzt werden und modifiziert werden.

```
\renewbibmacro{<name>}[<arguments>][<optional>]{<definition>}
\renewbibmacro*{<name>}[<arguments>][<optional>]{<definition>}
```

Ähnlich wie `\newbibmacro`, definiert jedoch `<name>` neu. Im Gegensatz zu `\renewcommand` bringt `\renewbibmacro` eine Warnmeldung, wenn das Makro undefiniert ist und fällt in `\newbibmacro` zurück.

```
\providebibmacro{<name>}[<arguments>][<optional>]{<definition>}
\providebibmacro*{<name>}[<arguments>][<optional>]{<definition>}
```

Ähnlich wie `\newbibmacro`, definiert `<name>` aber nur dann, wenn er undefiniert ist. Dieser Befehl ist vom Konzept her ähnlich wie `\providecommand`.

```
\usebibmacro{<name>}
```

Führt das Makro `<name>` aus, wie es mit `\newbibmacro` definiert wurde. Falls das Makro irgendwelche Argumente aufnimmt, dann werden Sie einfach nach `<name>` hinzugefügt. `\usebibmacro` ist robust.

```
\savecommand{<command>}
\restorecommand{<command>}
```

Diese Befehle sichern und stellen jeden Befehl wieder her, welcher ein Befehlsname sein muss, der mit einem Backslash anfängt. Beide Befehle arbeiten in einem lokalen Bereich. Sie werden hauptsächlich für die Verwendung in Lokalisierungsdateien zur Verfügung gestellt.

```
\savebibmacro{<name>}
\restorebibmacro{<name>}
```

Diese Befehle sichern und stellen das Makro `<name>` wieder her, wobei `<name>` ein Bezeichner für ein Makro ist, welches mit `\newbibmacro` definiert wurde. Beide Befehle arbeiten in einem lokalen Bereich. Sie werden hauptsächlich für die Verwendung in Lokalisierungsdateien zur Verfügung gestellt.

```
\savefieldformat[<entry type>]{<format>}
\restorefieldformat[<entry type>]{<format>}
```

Diese Befehle sichern und stellen die Formatierungsrichtlinie `<format>` wieder her, wie Sie mit `\DeclareFieldFormat` definiert wurde. Beide Befehle arbeiten innerhalb eines lokalen Bereichs. Sie sind hauptsächlich für die Verwendung in Lokalisierungsdateien vorgesehen.

```
\savelistformat[⟨entry type⟩]{⟨format⟩}
\restorelistformat[⟨entry type⟩]{⟨format⟩}
```

Diese Befehle sichern und stellen die Formatierungsrichtlinie *⟨format⟩* wieder her, wie Sie mit `\DeclareListFormat` definiert wurde. Beide Befehle arbeiten innerhalb eines lokalen Bereichs. Sie sind hauptsächlich für die Verwendung in Lokalisierungsdateien vorgesehen.

```
\savenameformat[⟨entry type⟩]{⟨format⟩}
\restorenameformat[⟨entry type⟩]{⟨format⟩}
```

Diese Befehle sichern und stellen die Formatierungsrichtlinie *⟨format⟩* wieder her, wie Sie mit `\DeclareNameFormat` definiert wurde. Beide Befehle arbeiten innerhalb eines lokalen Bereichs. Sie sind hauptsächlich für die Verwendung in Lokalisierungsdateien vorgesehen.

```
\usedriver{⟨code⟩}{⟨entrytype⟩}
```

Führt die Bibliographietreiber für einen *⟨entrytype⟩* aus. Der Aufruf dieses Befehls in dem *⟨loopcode⟩* eines Zitierbefehls, der mit `\DeclareCiteCommand` definiert wurde, ist ein einfacher Weg, komplette Zitate ähnlich einem Bibliographieeintrag zu drucken. Befehle wie `\newblock`, die in einem Zitat nicht anwendbar sind, werden automatisch deaktiviert. Zusätzliche Initialisierungsbefehle können als das *⟨code⟩*-Argument übergeben werden. Dieses Argument wird innerhalb der Gruppe, in der `\usedriver` den jeweiligen Treiber startet, ausgeführt. Beachten Sie, dass er in Hinblick auf den Syntax zwar erforderlich ist, jedoch auch leer bleiben kann und dass dieser Befehl automatisch die Sprache wechselt, wenn die Paketoption `babel` aktiviert ist.

```
\bibhypertarget{⟨name⟩}{⟨text⟩}
```

Ein Packer für `hyperrefs` `\hypertarget`-Befehl. *⟨name⟩* ist der Name des Fixpunkt, *⟨text⟩* ist ein beliebig druckbarer Text oder Kode, welcher als Fixpunkt dient. Gibt es in diesem Dokument irgendwelche `refsection`-Umgebungen, dann ist der *⟨name⟩* lokal für die aktuelle Umgebung. Ist die Package Option `hyperref` deaktiviert oder das `hyperref`-Paket wurde nicht geladen, dann übergibt dieser Befehl einfach das *⟨text⟩*-Argument. Siehe auch die Formatierungsrichtlinie `\bibhypertarget` in § 4.10.4.

```
\bibhyperlink{⟨name⟩}{⟨text⟩}
```

Ein Packer für `hyperrefs` `\hyperlink`-Befehl. *⟨name⟩* ist der Name eines Fixpunkts, der mit `\bibhypertarget` definiert wurde, *⟨text⟩* ist ein beliebig druckbarer Text oder Kode, der in einen Link umgewandelt werden soll. Gibt es in diesem Dokument irgendwelche `refsection`-Umgebungen, dann ist *⟨name⟩* lokal für die aktuelle Umgebung. Ist die Paketoption `hyperref` deaktiviert oder wurde das `hyperref`-Paket

nicht geladen, dann übergibt dieser Befehl einfach das  $\langle text \rangle$ -Argument. Siehe auch die Formatierungsrichtlinie `bibhyperlink` in § 4.10.4.

`\bibhyperref` $\{ \langle entrykey \rangle \} \{ \langle text \rangle \}$

Wandelt  $\langle text \rangle$  in einen internen Link um, der auf  $\langle entrykey \rangle$  in der Bibliographie zeigt. Entfällt der  $\langle entrykey \rangle$ , dann benutzt dieser Befehl den Schlüssel des Eintrags, der gerade bearbeitet wird. Dieser Befehl wird verwendet, um Zitate in anklickbare Links umzuwandeln, die auf den entsprechenden Eintrag in der Bibliographie zeigen. Das Linkziel wird von `biblatex` automatisch markiert. Gibt es in diesem Dokument mehrere Bibliographien, dann ist das Ziel das erste Auftreten von  $\langle entrykey \rangle$  in einer der Bibliographien. Gibt es irgendwelche `refsection`-Umgebungen, dann sind die Links lokal in dieser Umgebung. Siehe auch die Formatierungsrichtlinien `bibhyperref` in § 4.10.4.

`\ifhyperref` $\{ \langle true \rangle \} \{ \langle false \rangle \}$

Erweitert um  $\langle true \rangle$ , falls die Paketoption `hyperref` aktiviert ist (was bedeutet, dass das `hyperref`-Paket geladen wurde), und ansonsten um  $\langle false \rangle$ .

`\docsvfield` $\{ \langle field \rangle \}$

Ähnlich wie der `\docsvlist`-Befehl aus dem `etoolbox`-Package, außer dass er den Namen eines Feldes als Argument nimmt. Der Wert dieses Feldes ist wie eine durch Komma-getrennte Liste gegliedert. Ist das  $\langle field \rangle$  undefiniert, dann erweitert dieser Befehl um eine leere Zeichenkette.

`\forcsvfield` $\{ \langle handler \rangle \} \{ \langle field \rangle \}$

Ähnlich wie der `\forcsvlist`-Befehl aus dem `etoolbox`-Paket, außer dass er den Namen eines Feldes als Argument nimmt. Der Wert dieses Feldes ist wie eine durch Komma-getrennte Liste gegliedert. Ist das  $\langle field \rangle$  undefiniert, dann erweitert dieser Befehl um eine leere Zeichenkette.

`\MakeCapital` $\{ \langle text \rangle \}$

Ähnlich wie `\MakeUppercase`, wandelt jedoch nur den ersten druckbaren Buchstaben von  $\langle text \rangle$  in einen Großbuchstaben um. Beachte, dass die Einschränkungen, die für `\MakeUppercase` gelten, auch für diesen Befehl gelten. Es müssen nämlich alle Befehle in  $\langle text \rangle$  robust sein oder müssen `\protect` vorangestellt haben, da  $\langle text \rangle$  während der Erstellung der Überschrift erweitert ist. Abgesehen von ASCII-Zeichen und Standard-Akzent-Befehlen arbeitet der Befehl auch mit aktiven Zeichen des `inputenc`-Paket, genauso wie mit Kurzschriften des `babel`-Pakets. Startet  $\langle text \rangle$  mit einer Kontrollsequenz, dann wird nichts groß geschrieben. Dies ist ein robuster Befehl.

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

Konvertiert das *<text>*-Argument in die Satzschreibung, z. B. wird das erste Wort groß geschrieben und der Rest der Kette klein. Dies ist ein robuster Befehl. Die Variante mit dem Asterisk unterscheidet sich von der regulären Version darin, dass Sie die Sprache des Eintrags betrachtet, was in dem `hyphenation`-Feld angegeben ist. Er konvertiert nur den *<text>* in die Satzschreibung, wenn das `hyphenation`-Feld undefiniert ist oder eine Sprache enthält, die mit `\DeclareCaseLangs` deklariert wurde (siehe unten).<sup>31</sup> Ansonsten ist *<text>* in keiner Weise verändert. Es wird in den Formatierungsrichtlinien empfohlen, `\MakeSentenceCase*` anstatt der regulären Variante zu verwenden. Beide Varianten unterstützen die traditionelle Abmachung für `bib`-Dateien, dass alles, was von geschweiften Klammern umgeben ist, nicht verändert wird, wenn die Schreibung verändert wird. Zum Beispiel:

```
\MakeSentenceCase{an Introduction to LaTeX}
\MakeSentenceCase{an Introduction to {LaTeX}}
```

Würde ergeben:

```
An introduction to latex An introduction to LaTeX
```

In `bib`-Dateien wird mit dem traditionellen BibTeX geformt in dem Sinn, dass es recht gängig ist, nur einzelne Buchstaben in Klammern zu setzen, um eine Veränderung der Schreibung zu verhindern:

```
title = {An Introduction to {L}a{T}e{X}}
```

Das Problem mit dieser Konvention ist, dass die Klammern das Kerning auf die an beiden Seiten anliegenden Buchstaben unterdrücken. Es ist besser, das ganze Wort in geschweiften Klammern, wie im ersten Beispiel gezeigt, zu packen.

```
\mkpageprefix[<pagination>][<postpro>]{<text>}
```

Dieser Befehl ist für die Verwendung in Feldformatierungsrichtlinien bestimmt, welche die Seitennummern in dem *<postnote>*-Argument von Zitierbefehlen und dem `pages`-Feld von Bibliographieeinträgen formatieren. Er analysiert das *<text>*-Argument und stellt standardmäßig ‚p.‘ oder ‚pp.‘ voran. Das optionale *<pagination>*-Argument beinhaltet den Namen eines Feldes, welches den Nummerierungstyp angibt. Dies kann entweder `pagination` oder `bookpagination` sein, wobei `pagination`

<sup>31</sup>Die Konvertierung in die Satzschreibung ist für folgenden Sprachkennungen standardmäßig aktiviert: `american`, `british`, `canadian`, `english`, `australian`, `newzealand` genauso wie der Deckname `USenglish` und `UKenglish`

der Standardwert ist. Der Platz zwischen dem Präfix und dem  $\langle text \rangle$  kann durch die Neudefinition von `\ppspace` verändert werden. Siehen Sie §§ 2.3.10 und 3.11.3 für mehr Details. Siehe auch `\DeclareNumChars`, `\DeclareRangeChars`, `\DeclareRangeCommands`, und `\NumCheckSetup`. Das optionale  $\langle postpro \rangle$ -Argument bestimmt ein Makro, welches für die Nachbereitung von  $\langle text \rangle$  benutzt wird. Dies muss ein Makro sein, welches zwingend ein Makro aufnimmt. Falls nur ein optionales Argument gegeben ist, dann wird es als  $\langle pagination \rangle$  genommen. Dazu zwei typische Beispiele:

```
\DeclareFieldFormat{postnote}{\mkpageprefix[pagination]{#1}}
\DeclareFieldFormat{pages}{\mkpageprefix[bookpagination]{#1}}
```

Das optionale Argument `pagination` kann in dem ersten Beispiel weggelassen werden.

`\mkpagetotal` [ $\langle pagination \rangle$ ] [ $\langle postpro \rangle$ ] { $\langle text \rangle$ }

Dieser Befehl ist ähnlich wie `\mkpageprefix`, außer das er für das `pagetotal` Feld von Bibliographieinträgen bestimmt ist, z. B. würde er „123 pages“ anstatt „page 123“ drucken. Ist das optionale  $\langle pagination \rangle$ -Argument nicht vorhanden, so ist wird `bookpagination`-Argument automatisch gesetzt. Der Abstand, der zwischen dem Nummerierungssuffix und dem  $\langle text \rangle$  eingefügt wurde, kann durch die Neudefinierung des `\ppspace`-Makros verändert werden. Das optionale  $\langle postpro \rangle$ -Argument bestimmt ein Makro, welches für die Nachbearbeitung von  $\langle text \rangle$  benutzt wird. Dies muss ein Makro sein, welches zwingend ein Argument beinhaltet. Ist nur ein Argument gegeben, dann wird es als  $\langle pagination \rangle$  genutzt. Dazu ein typisches Beispiel:

```
\DeclareFieldFormat{pagetotal}{\mkpagetotal[bookpagination]{#1}}
```

Das optionale Argument `bookpagination` kann in diesem Fall weggelassen werden.

`\mkcomprange` [ $\langle postpro \rangle$ ] { $\langle text \rangle$ }

Dieser Befehl, der für Richtlinien der Feldformatierung vorgesehen ist, analysiert das  $\langle text \rangle$ -Argument für Seitenbereiche und komprimiert es. Beispielsweise kann „125–129“ formatiert sein als „125–9“. Sie können das Verhalten von `\mkcomprange` durch das Anpassen der LaTeX-Zähler `mincomprange`, `maxcomprange` und `mincompwidth` anpassen, wie in table 8 dargestellt. Die Standardeinstellungen sind 10, 100000 bzw. 1. Dies bedeutet, dass der Befehl versucht, standardmäßig so viel wie möglich zu komprimieren, nehmend `\setcounter`, um die Parameter einzustellen. Der Scanner erkennt `\bibrangedash` und Bindestriche als Bereichstriche. Es wird der Bindestrich mittels Ersetzen einer beliebigen Anzahl von aufeinander folgenden Trennstriche mit `\bibrangedash` normalisiert. Die Liste der Trennungen mit Kommatas und/oder Semikolons wird ebenfalls unterstützt. Wenn Sie ein Zeichen aus der Liste/Kette aus irgendeinem Grund verstecken möchten, setzen Sie den Buchstaben

| Input     | Output                         |                               |                                |
|-----------|--------------------------------|-------------------------------|--------------------------------|
|           | <code>mincomprange=10</code>   | <code>mincomprange=100</code> | <code>mincomprange=1000</code> |
| 11-15     | 11-5                           | 11-15                         | 11-15                          |
| 111-115   | 111-5                          | 111-5                         | 111-115                        |
| 1111-1115 | 1111-5                         | 1111-5                        | 1111-5                         |
|           | <code>maxcomprange=1000</code> | <code>maxcomprange=100</code> | <code>maxcomprange=10</code>   |
| 1111-1115 | 1111-5                         | 1111-5                        | 1111-5                         |
| 1111-1155 | 1111-55                        | 1111-55                       | 1111-1155                      |
| 1111-1555 | 1111-555                       | 1111-1555                     | 1111-1555                      |
|           | <code>mincompwidth=1</code>    | <code>mincompwidth=10</code>  | <code>mincompwidth=100</code>  |
| 1111-1115 | 1111-5                         | 1111-15                       | 1111-115                       |
| 1111-1155 | 1111-55                        | 1111-55                       | 1111-155                       |
| 1111-1555 | 1111-555                       | 1111-555                      | 1111-555                       |

Tabelle 8: `\mkcomprange`-Aufbau

oder die gesamte Zeichenfolge in geschweifte Klammern. Das optionale `<postpro>`-Argument legt ein Makro zur Nachbearbeitung von `<text>` fest. Dies ist wichtig, wenn Sie `\mkcomprange` mit anderen Formatierungsmakros verbinden möchten, die Sie auch benötigen, um ihr `<text>`-Argument, so eins wie `\mkpageprefix`, zu parsen. Einfache Serien dieser Befehle arbeiten nicht wie erwartet. Verwenden Sie das `<postpro>`-Argument für das Einrichten der Prozesskette wie folgt:

```
\DeclareFieldFormat{postnote}{\mkcomprange[{\mkpageprefix[pagination]}]{#1}}
```

Beachten Sie, dass `\mkcomprange` zuerst ausgeführt wird, nehmend `\mkpageprefix` als Nach-Prozessor. Berücksichtigen Sie auch, dass das `<postpro>`-Argument in ein zusätzliches Paar von Klammern eingeschlossen ist. Dies ist nur in einem speziellen Fall erforderlich, um zu verhindern, dass LaTeX's optionaler Argumentscanner die vorherigen verschachtelten Klammern verwechselt.

```
\mkpagefirst[<postpro>]{<text>}
```

Dieser Befehl, welcher für die Benutzung in Feldformatierungsrichtlinien bestimmt ist, zergliedert das `<text>`-Argument in Seitenbereiche und druckt die Startseite des Bereiches. Die reguläre Version benutzt `\mkpageprefix`, um den `<text>` nach der Kürzung zu formatieren, welcher das `<pagination>` und/oder das `<postpro>`-Argument an `\mkpageprefix` übergibt. Die Variante mit dem Asterisk druckt direkt das gekürzte `<text>`-Argument. Das optionale `<postpro>`-Argument bestimmt ein Makro, welches für die Nachbearbeitung von `<text>` benutzt wird. Dieses Makro muss zwingend ein Argument enthalten.

```
\DeclareNumChars{<characters>}
```

```
\DeclareNumChars*{<characters>}
```

Dieser Befehl konfiguriert die `\ifnumeral`, `\ifnumerals` und `\ifpages` Tests aus § 4.6.2. Das Setup beeinflusst auch `\iffieldnum`, `\iffieldnums` und `\iffieldpages`, genauso wie `\mkpageprefix` und `\mkpagetotal`. Das `<characters>`-Argument ist eine unbegrenzte Liste von Zeichen, die als Teil einer Zahl betrachtet werden. Die reguläre Version dieses Befehls ersetzt die aktuelle Einstellung, die Variante mit dem Asterisk hängt die Argument an die aktuelle Liste an. Die Standardeinstellung ist:

```
\DeclareNumChars{.}
```

Dies bedeutet, dass eine Nummer (von einem Abschnitt oder ähnlichem) wie ‚3.4.5‘ als eine Zahl angesehen wird. Beachten Sie, dass arabische und römische Zahlen standardmäßig erkannt werden, weshalb Sie nicht explizit deklariert werden müssen.

```
\DeclareRangeChars{<characters>}
```

```
\DeclareRangeChars*{<characters>}
```

Dieser Befehl konfiguriert die `\ifnumerals`- und `\ifpages`-Tests aus § 4.6.2. Das Setup beeinflusst auch `\iffieldnums` und `\iffieldpages`, genauso wie `\mkpageprefix` und `\mkpagetotal`. Das `<characters>`-Argument ist eine unbegrenzte Liste von Zeichen, die als Abschnittsindikatoren betrachtet werden. Die reguläre Variante dieses Befehls ersetzt die aktuelle Einstellung, die Variante mit dem Asterisk hängt die Argument an die aktuelle Liste. Die Standardeinstellung ist:

```
\DeclareRangeChars{~,;,+/-}
```

Das bedeutet, dass Zeichenketten wie ‚3–5‘, ‚35+‘, ‚8/9‘, usw. von `\ifnumerals` und `\ifpages` als ein Abschnitt betrachtet werden. Siehe auch §§ 2.3.10 und 3.11.3 für mehr Details.

```
\DeclareRangeCommands{<commands>}
```

```
\DeclareRangeCommands*{<commands>}
```

Dieser Befehl ist ähnlich wie `\DeclareRangeChars`, außer dass das `<commands>`-Argument eine unbegrenzte Liste von Befehlen ist, welche als Abschnittsindikatoren betrachtet werden. Die reguläre Variante dieses Befehls ersetzt die aktuelle Einstellung, die Variante mit dem Asterisk hängt die Argument an die aktuelle Liste. Die Standardliste ist ziemlich lang und sollte alle üblichen Fälle abdecken. Dazu ein kurzes Beispiel:

```
\DeclareRangeCommands{\&\bibrangedash\textendash\textemdash\psq\psqq}
```

Siehe auch §§ 2.3.10 und 3.11.3 für mehr Details.

```
\DeclarePageCommands{⟨commands⟩}
\DeclarePageCommands*{⟨commands⟩}
```

Dieser Befehl ist ähnlich wie `\DeclareRangeCommands`, außer dass er nur die `\ifpages-` und `\iffieldpages-`Tests beeinflusst, aber nicht `\ifnumerals` und `\iffieldnums`. Die Standardeinstellung ist:

```
\DeclarePageCommands{\pno\ppno}
```

```
\NumCheckSetup{⟨code⟩}
```

Benutzen Sie diesen Befehl, um irgendwelche Befehle neu zu definieren, der die Tests, die von `\ifnumeral`, `\ifnumerals`, und `\ifpages` aus § 4.6.2 ausgeführt werden, beeinträchtigt. Das Setup beeinflusst auch `\iffieldnum`, `\iffieldnums` und `\iffieldpages`, genauso wie `\mkpageprefix` and `\mkpagetotal`. Der `⟨code⟩` wird von einer Gruppe dieser Befehle ausgeführt. Da die oben genannten Befehle den String erweitern, um analysiert zu werden, ist es möglich, dass Sie von den Tests ignoriert werden, indem man sie zu einem leeren String erweitern lässt. Siehe auch §§ 2.3.10 and 3.11.3 für mehr Details.

```
\DeclareCaseLangs{⟨languages⟩}
\DeclareCaseLangs*{⟨languages⟩}
```

Definiert die Liste der Sprachen, die von dem `\MakeSentenceCase*`-Befehl betrachtet werden, der einen String in die Satzschreibung umwandelt. Das `⟨languages⟩`-Argument ist eine durch Komma-getrennte Liste von babel Sprachindikatoren. Die reguläre Version dieses Befehls ersetzt die aktuelle Einstellung, die Version mit dem Asterisk hängt das Argument an die aktuelle Liste. Die Standardeinstellung ist:

```
\DeclareCaseLangs{%
american,british,canadian,english,australian,newzealand, USenglish,UKenglish}
```

Siehe das babel-Handbuch und Tabelle 1 für eine Liste der Sprachindikatoren.

```
\BibliographyWarning{⟨message⟩}
```

Dieser Befehl ist ähnlich wie `\PackageWarning`, druckt jedoch zusätzlich zur Eingabezeilennummer den Schlüssel des Eintrags, der gerade bearbeitet wird. Er kann in einer Bibliographie sowie in Zitierbefehlen verwendet werden. Ist die `⟨message⟩` ziemlich lang, dann benutzen Sie `\MessageBreak`, um Zeilenumbrüche einzufügen. Beachten Sie, dass der Standardbefehl `\PackageWarning` keinen aussagekräftigen Hinweis bereitstellt, wenn der Befehl in einer Bibliographie benutzt wird, da die Zeilennummer der Eingabe die Zeile angibt, in der der `\printbibliography` Befehl gegeben war.

`\RequireBiber` [*severity*]

Dieser Befehl ist für die Verwendung in `cbx/bbx`-Dateien und in der `@preamble` von `bib`-Dateien bestimmt. Er überprüft das ausgewählte Backend und warnt, wenn es nicht Biber ist. Das optionale *severity*-Argument ist ein Integer, der die Gewichtung bestimmt. Der Wert 1 bewirkt eine Information, die mitteilt, dass Biber empfohlen wird; 2 bewirkt eine Warnung, die mitteilt, dass Biber benötigt wird und die `Stil/bib`-Datei nicht richtig funktionieren könnte; 3 bewirkt eine Fehleranzeige, die mitteilt, dass Biber unbedingt benötigt wird und dass die `Stil/bib`-Datei mit keinem anderen Backend funktionieren wird. Wird `\RequireBiber` mehrmals benutzt, dann erhält die höchste Gewichtung Vorrang. `cbx/bbx`-Dateien auf der einen Seite und die `@preamble`-Ausschnitte aller `bib`-Dateien auf der anderen Seite werden separat verfolgt. Falls das optionale *severity*-Argument fehlt, dann ist die Standardgewichtung 2 (Warnung).

`\pagetrackertrue`  
`\pagetrackerfalse`

Dieser Befehl aktiviert oder deaktiviert die Zitatverfolgung lokal (dies wirkt sich auf die `\iffirstonpage`- und `\ifsamepage`- Tests aus (§ 4.6.2)). Sie sind für die Verwendung in der Definition eines Zitierbefehls oder irgendwo im Hauptteil des Dokuments bestimmt. Wird ein Zitierbefehl von der Seitenverfolgung ausgeschlossen, dann benutze `\pagetrackerfalse` in dem *precode*-Argument von `\DeclareCiteCommand`. Siehe § 4.3.1 für Details. Beachten Sie, dass dieser Befehl nichts bewirkt, wenn die Seitenverfolgung global deaktiviert wurde.

`\citetrackertrue`  
`\citetrackerfalse`

Diese Befehle aktivieren oder deaktivieren alle Zitatverfolger lokal (dies wirkt sich auf die `\ifciteseen`-, `\ifentryseen`-, `\ifciteibid`- und `\ifciteidem`-Tests aus (§ 4.6.2)). Sie sind für die Verwendung in Zitierbefehlen oder irgendwo in im Hauptteil des Dokuments bestimmt. Wird ein Zitierbefehl von der Verfolgung ausgeschlossen, dann benutze `\citetrackerfalse` in dem *precode*-Argument von `\DeclareCiteCommand`. Siehe § 4.3.1 für Details. Beachten Sie, dass dieser Befehl nichts bewirkt, wenn die Verfolgung global deaktiviert wurde.

`\backtrackertrue`  
`\backtrackerfalse`

Diese Befehle aktivieren oder deaktivieren den `backref`-Verfolger lokal. Sie sind für die Verwendung in Zitierbefehlen oder irgendwo im Hauptteil des Dokuments bestimmt. Wird ein Zitierbefehl von Rückverfolgung ausgeschlossen, dann benutze `\backtrackerfalse` in dem *precode*-Argument vom `\DeclareCiteCommand`. Beachten Sie, dass diese Befehle nichts bewirken, wenn die `backref`-Option nicht global gesetzt wurde.

## 4.7. Zeichensetzung & Abstände

Das `biblatex`-Paket stellt ausführliche Hilfsmittel zur Verfügung, die für die Verwaltung und Nachverfolgung von Satzzeichen und Leerräumen in Bibliographien und

Zitaten konzipiert sind. Diese Hilfsmittel arbeiten auf zwei Ebenen. Die Befehle der oberen Ebene, die in § 4.7.1 erklärt wurden, arbeiten mit Satz- und Leerzeichen, die von dem Bibliographiestil zwischen den verschiedenen Abschnitten eines Bibliographieeintrags eingefügt werden. Die Befehle in §§ 4.7.2, 4.7.3, 4.7.4 arbeiten auf der niedrigeren Ebene. Sie nutzen TeX's Füllfaktor und veränderte Füllfaktoren, um die Zeichensetzung auf eine stabile und effiziente Art und Weise zu verfolgen. Auf diesem Weg ist es möglich, angehängte Satzzeichen innerhalb von Feldern zu erkennen und nicht nur die Satzzeichen, die genau zwischen Feldern eingefügt wurden. Die selbe Technik wird auch für die automatische Großschreibung von Lokalisierungsstrings benutzt. Siehe dazu `\DeclareCapitalPunctuation` in § 4.7.5 und § 4.8 für Details. Beachte, dass diese Hilfsmittel nur lokal in Zitaten und Bibliographien zur Verfügung stehen. Sie beeinflussen keine anderen Teile des Dokuments.

#### 4.7.1. Block- und Unit/Einheit-Interpunktion

Die Hauptsegmente eines Bibliographieeintrags sind ‚blocks‘ und ‚units‘. Ein Block ist das größere Segment von beiden, ein Unit (eine Einheit) das kürzere oder höchstens gleich lange. Beispielsweise bilden die Werte der Felder, wie `title` oder `note`, üblicherweise eine Einheit, die von den folgenden Daten durch einen Punkt oder ein Komma getrennt wird. Ein Block kann verschiedene Felder umfassen, die als separate Einheiten behandelt werden, beispielsweise `publisher`, `location` und `year`. Die Segmentierung eines Eintrags in Blöcke und Einheiten legt im Ermessen der Bibliographiestile. Ein Eintrag wird segmentiert durch das Einführen der Befehle `\newblock` und `\newunit` an geeigneten Stellen und des `\finentry`-Befehl ganz am Ende (sehen Sie in § 4.2.3 ein Beispiel). Schauen Sie auch in § 4.11.7 für weitere praktische Hinweise.

- `\newblock` Verzeichnet das Ende eines Blocks. Dieser Befehl druckt nicht alles, er markiert nur das Blockende. Der Blockabgrenzer `\newblockpunct` wird durch einen abschließenden `\printtext-`, `\printfield-`, `\printlist-`, `\printnames-` oder `\bibstring-` Befehl eingefügt. Sie können `\newblock` an geeigneten Plätzen nutzen, ohne sich um falsche Blöcke zu kümmern. Ein neuer Block wird nur vom nächsten `\printfield-` (oder ähnlichem) Befehl gestartet werden, falls dieses Kommando nichts druckt. Zu weiteren Details sehen Sie § 4.11.7.
- `\newunit` Aufzeichnungen des Endes einer Einheit und erstellen der Standardtrennzeichen `\newunitpunct` im Zeichensetzungs-puffer. Dieser Befehl druckt nicht alles, sondern markiert lediglich das Einheitenende. Der Zeichensetzung-Puffer wird mit dem nächsten `\printtext-`, `\printfield-`, `\printlist-`, `\printnames-` oder `\bibstring-` Befehl eingefügt. Sie können `\newunit` nach Befehlen wie `\printfield` nehmen, ohne sich um eine falsche Interpunktion und Leerzeichen zu kümmern. Der Puffer wird nur bis zum nächsten `\printfield-` oder einem ähnlichen Befehl einge-

fügt, wenn *beide* Felder nicht leer sind. Dies gilt auch für `\printtext`, `\printlist`, `\printnames` und `\bibstring`. Zu weiteren Details sehen Sie § 4.11.7.

`\finentry` Fügt `\finentrypunct` ein. Dieser Befehl sollte am Ende eines jeden Bibliographie-eintrages verwendet werden.

```
\setunit{<punctuation>}
\setunit*{<punctuation>}
```

Der `\setunit`-Befehl ähnelt `\newunit`, außer dass er `<punctuation>` (Satzzeichen) verwendet anstelle von `\newunitpunct`. Die gesternte Variante unterscheidet sich von der regulären Form, indem sie prüft, ob der letzte `\printtext`-, `\printfield`-, `\printlist`-, `\printnames`- oder `\bibstring`-Befehl tatsächlich nichts gedruckt hat. Wenn nicht, bedeutet es nichts.

```
\setpunctfont{<command>}
```

Dieser Befehl, der für Richtlinien zur Feldformatierung vorgesehen ist, ist ein alternativer Weg für den Umgang mit Einheitenzeichensetzung, wenn ein Feld in einer anderen Schriftart gedruckt ist (beispielsweise ein Titel kursiv). Standardmäßig geht LaTeX damit um, indem es eine kleine Menge an Speicherplatz zugibt, die sogenannte Kursiv-Korrektur. Dieser Befehl ermöglicht die Anpassung der Interpunktion an die Schriftart des Vorgängerfelds. Der `<command>`-Befehl sollte ein Schriftartbefehl sein, der sein Argument so wie `\emph` oder `\textbf` nimmt. Dieser Befehl wirkt sich nur auf Satzzeichen mit den Befehlen in § 4.7.3 aus. Die Anpassung der Schriftart wird nur auf die nächsten Satzzeichen angewendet und danach wird sie wieder zurückgesetzt. Wenn Sie möchten, können sie manuell zurücksetzen, bevor dieser Effekt eintritt, nehmen Sie dazu `\resetpunctfont`. Wenn die `punctfont`-Paketoption deaktiviert ist, arbeitet dieser Befehl nicht. Beachten Sie, dass die `\mkbibemph` und `\mkbibbold` Wrappers (Hüllen) aus § 4.10.4 diese Funktion standardmäßig integrieren.

`\resetpunctfont` Dieser Befehl deaktiviert die Schriftdefinitionen zur Einheitenzeichensetzung mit `\setpunctfont`, bevor sie wirksam werden. Wenn die `punctfont`-Paketoption deaktiviert ist, funktioniert dieser Befehl nicht.

#### 4.7.2. Interpunktionstests

Die folgenden Befehle können verwendet werden, um die letzten Satzzeichen an jedem beliebigen Punkt in Zitaten und Literaturquellen zu testen.

```
\ifpunct{<true>}{<>false>}
```

Nimmt `<true>` für alle Satzzeichen, außer es geht ein Punkt voraus und ansonsten `<>false>`.

`\ifterm{⟨true⟩}{⟨false⟩}`

Führt *⟨true⟩* aus, wenn ein terminales Satzzeichen vorausgeht, ansonsten *⟨false⟩*. Ein terminales Satzzeichen ist ein Satzzeichen, das für die automatische Großschreibung registriert ist, entweder mit `\DeclareCapitalPunctuation` oder als Standard, in § 4.7.5 finden Sie Details. Standardmäßig gilt dies für Ausrufungs- und Fragezeichen.

`\ifpunctmark{⟨character⟩}{⟨true⟩}{⟨false⟩}`

Nimmt *⟨true⟩*, wenn das Satzzeichen *⟨character⟩* vorausgeht, ansonsten *⟨false⟩*. *⟨character⟩* kann ein Komma, ein Semikolon, ein Doppelpunkt, ein Punkt, ein Ausrufungszeichen, ein Fragezeichen oder ein Sternchen sein. Beachten Sie, dass ein Punkt das Ende eines Satzes kennzeichnet. Verwenden Sie ein Sternchen, um den Punkt nach einer Abkürzung zu testen. Wenn dieser Befehl in einer Richtlinie zur Namensformatierung verwendet wird, d. h. im Argument von `\DeclareNameFormat`, kann *⟨character⟩* auch ein Apostroph sein.

### 4.7.3. Hinzufügung von Satzzeichen

Die folgenden Befehle wurden so ausgelegt, dass sie eine Dopplung der Satzzeichen verhindern. Literaturnachweis- und Zitierungsstile sollten diese Befehle immer benutzen, anstelle der eigentlichen (literal) Satzzeichen. Alle `\add...-`Befehle in diesem Abschnitt entfernen automatisch vorangehende Leerzeichen mit `\unspace` (sehen Sie § 4.7.4). Beachten Sie, dass das Verhalten aller `\add...-`Befehle besprochen wird, unten in den Paketstandards, die wiederhergestellt werden, wenn `biblatex` die Sprachen wechselt. Dieses Verhalten kann mit `\DeclarePunctuationPairs` aus § 4.7.5 eingestellt werden.

`\adddot` Fügt einen Punkt ein, wenn es vorangestellt ist einem Satzzeichen. Der Zweck dieses Befehls ist das Einfügen eines Punktes nach einer Abkürzung. Jeder Punkt, der in dieser Art eingefügt wird, wird von den anderen Interpunktionsbefehlen als solcher erkannt. Dieser Befehl kann auch verwendet werden, um vorher eingefügte eigentliche Punkte in einen Abkürzungspunkt umzuwandeln.

`\addcomma` Fügt ein Komma ein, wenn ein anderes Komma, ein Semikolon, ein Doppelpunkt oder ein Punkt vorangestellt ist.

`\addsemicolon` Fügt ein Semikolon ein, wenn ein Komma, ein anderes Semikolon, ein Doppelpunkt oder ein Punkt vorangestellt ist.

`\addcolon` Fügt einen Doppelpunkt ein, wenn ein Komma, ein Semikolon, ein anderer Doppelpunkt oder ein Punkt vorangestellt ist.

- `\addperiod` Fügt einen Punkt ein, wenn ein anderer Punkt oder ein anderes Satzzeichen vorangestellt ist. Dieses Kommando kann auch verwendet werden, um einen zuvor angebrachten Abkürzungspunkt in einen Punkt umzuwandeln, beispielsweise am Satzende.
- `\addexclam` Fügt ein Ausrufungszeichenein, wenn es vorausgeht einem Satzzeichen, ausgeschlossen ist ein Abkürzungspunkt.
- `\addquestion` Fügt ein Fragezeichen ein, wenn es vorausgeht einem Punkt, ausgeschlossen ist ein Abkürzungspunkt.
- `\isdot` Verwandelt einen zuvor eingebrachten eigentlichen Punkt in einen Abkürzungspunkt. Im Gegensatz zu `\addot` ist nichts eingefügt, wenn dieser Befehl nicht einem Punkt vorausgeht.
- `\nopunct` Fügt einen internen Marker ein, der dazu führt, dass der nächste Satzzeichenbefehl nicht gedruckt wird.

#### 4.7.4. Hinzufügen von Leerzeichen

Die folgenden Befehle wurden entwickelt, um störende Leerzeichen zu verhindern. Bibliographie- und Zitierstile sollten diese Befehle immer verwenden, anstelle von eigentlichen Leerzeichen. Im Gegensatz zu den Befehlen in §§ 4.7.2 und 4.7.3 sind sie nicht auf Zitate und das Literaturverzeichnis beschränkt, sondern global verfügbar.

- `\unspace` Entfernt vorangegangene Leerzeichen, d. h. es entfernt alle Sprünge (*skips*) und Strafpunkte (*penalties*) vom Ende der laufenden horizontalen Liste. Der Befehl wird implizit durch alle folgenden Befehle ausgeführt.
- `\addspace` Fügt einen zerbrechlichen wortinternen Leerraum ein.
- `\addnbspace` Fügt einen nicht zerbrechlichen wortinternen Leerraum ein.
- `\addthinspace` Fügt ein *zerbrechliches* schmales Leerzeichen (*thin space*) ein.
- `\addnbthinspace` Fügt einen nicht zerbrechliches schmales Leerzeichen ein. Dies ist ähnlich zu `\,` und `\thinspace`.
- `\addlowpenspace` Fügt ein Leerzeichen ein, bestraft mit dem Wert des `lownamepenalty`-Zählers, siehe §§ 3.8.3 und 4.10.3 für Details.
- `\addhighpenspace` Fügt ein Leerzeichen ein, bestraft mit dem Wert des `highnamepenalty`-Zählers, siehe §§ 3.8.3 und 4.10.3 für Details.
- `\addlpthinspace` Ähnlich zu `\addlowpenspace`, fügt aber einen zerbrechlichen schmalen Leerraum ein.

- `\addhpthinspace` Ähnlich zu `\addhighpensepace`, fügt aber einen zerbrechlichen schmalen Leerraum ein.
- `\addabbrvspace` Fügt ein Leerzeichen ein, bestraft mit dem Wert des `abbrvpenalty`-Zählers, siehe §§ 3.8.3 und 4.10.3 für Details.
- `\addabthinspace` Ähnlich zu `\addabbrvspace`, jedoch einen schmalen Leerraum nehmend.
- `\adddotsspace` Vollzieht `\adddot` und fügt ein Leerzeichen ein, bestraft mit dem Wert des `abbrvpenalty`-Zähler, siehe §§ 3.8.3 und 4.10.3 für Details.
- `\addslash` Fügt einen zerbrechlichen Schrägstrich (slash) ein. Dieser Befehl unterscheidet sich vom `\slash`-Befehl im LaTeX-Kernel, wo ein Zeilenumbruch hinter einem Slash überhaupt nicht bestraft wird.

Beachten Sie, dass die Befehle in diesem Abschnitt `\unspace` implizit ausführen, um störende Leerzeichen zu entfernen, daher können sie verwendet werden, um sich gegenseitig zu überschreiben. Beispielsweise können `\addnbspace` nehmen, um einen zuvor eingebrachtes wortinternes Leerzeichen in einen nicht zerbrechlichen zu verwandeln und `\addspace` verwandelt ein nicht zerbrechliches Leerzeichen in ein zerbrechliches.

#### 4.7.5. Konfiguration von Zeichensetzung und Großschreibung

Die folgenden Befehle konfigurieren verschiedene Funktionen im Zusammenhang mit Zeichensetzung und automatischer Großschreibung.

`\DeclareAutoPunctuation{⟨characters⟩}`

Dieser Befehl definiert, wie die Satzzeichen von den Zitierungsbefehlen behandelt werden, wenn diese den Text nach Satzzeichen durchsuchen. Beachten Sie, dass `⟨characters⟩` (Zeichen) keine unbegrenzte Liste von Zeichen darstellt. Gültige `⟨characters⟩` sind Punkt, Komma, Semikolon, Doppelpunkt, Ausrufezeichen und Fragezeichen. Die Standardeinstellung ist:

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

Diese Definition wird automatisch wiederhergestellt, sobald die `autopunct`-Paketoption auf `true` eingestellt ist. Den Befehl `\DeclareAutoPunctuation{}` auszuführen, ist äquivalent zur Einstellung `autopunct=false`, das bedeutet, er deaktiviert diese Funktion.

`\DeclareCapitalPunctuation{⟨characters⟩}`

Wenn biblatex lokalisierte Zeichenketten einsetzt, das heißt Schlüsselbegriffe wie ‚edition‘ (Ausgabe) oder ‚volume‘ (Umfang), schreibt es sie automatisch nach Zeichen, die einen Satz beenden, groß. Dieser Befehl definiert solche Satzzeichen, die bewirken, dass eingefügte Zeichenketten groß geschrieben werden, wenn eines dieser Zeichen davor steht. Beachten Sie, dass *⟨characters⟩* keine unbegrenzte Liste von Zeichen darstellt. Gültige *⟨characters⟩* sind Punkt, Komma, Semikolon, Doppelpunkt, Ausrufezeichen und Fragezeichen. Das Paket ist standardmäßig:

```
\DeclareCapitalPunctuation{.!?}
```

Die `\DeclareCapitalPunctuation` mit einem leeren Argument aufzurufen, ist äquivalent dazu, die automatische Großschreibung zu deaktivieren. Da diese Funktion sprachspezifisch ist, muss dieser Befehl in dem Argument von `\DefineBibliographyExtras` (falls er in der Präambel verwendet wird) oder `\DeclareBibliographyExtras` (falls er in einem lokalen Modul verwendet wird) verwendet werden. Betrachten Sie §§ 3.7 and 4.9 für weitere Details. Bei Verwendung der Standardeinstellung werden Zeichen nach Punkten, Ausrufezeichen und Fragezeichen groß geschrieben. Bei Beginn eines neuen Paragraphen werden alle Zeichenketten generell groß geschrieben (und zwar sobald TeX im vertikalen Modus ist).

`\DeclarePunctuationPairs{⟨identifier⟩}{⟨characters⟩}`

Benutzen Sie diesen Befehl, um gültige Paare von Satzzeichen zu deklarieren. Dies beeinflusst die Zeichensetzungsbefehle, die in § 4.7.3 diskutiert werden. Zum Beispiel besagt die Beschreibung von `\addcomma`, dass dieser Befehl ein Komma einfügt, wenn ein anderes Komma, ein Semikolon, ein Doppelpunkt oder Punkt vorausgeht. Mit anderen Worten sind Kommas nach Punkten einer Abkürzung, Ausrufezeichen und Fragezeichen verboten. Diese gültigen Paare werden wie folgt deklariert:

```
\DeclarePunctuationPairs{comma}{*!?!}
```

Der *⟨identifier⟩* (Bezeichner) wählt die Befehle aus, um sie zu konfigurieren. Die *⟨identifier⟩* entsprechen den Namen der Zeichensetzungsbefehle von § 4.7.3 ohne `\add` prefix. Das heißt, gültige *⟨identifier⟩*-Zeichen sind `dot` (Punkt), `comma` (Komma), `semicolon` (Semikolon), `colon` (Doppelpunkt), `period` (Punkt), `exclam` (Ausrufezeichen), `question` (Fragezeichen). Das Argument von *⟨characters⟩* ist eine nicht unbegrenzte Liste von Satzzeichen. Gültige *⟨characters⟩* sind Komma, Semikolon, Doppelpunkt, Punkt, Ausrufezeichen, Fragezeichen und Sternchen. Ein Punkt im Argument von *⟨characters⟩* steht für den Punkt am Ende eines Satzes, ein Sternchen für den Punkt nach einer Abkürzung. Dies ist die Standardeinstellung, die automatisch

wiederhergestellt wird, sobald biblatex die Sprache wechselt und den Einstellungen entspricht, wie sie in § 4.7.3 beschrieben sind:

```
\DeclarePunctuationPairs{dot}{}
\DeclarePunctuationPairs{comma}{*!?!}
\DeclarePunctuationPairs{semicolon}{*!?!}
\DeclarePunctuationPairs{colon}{*!?!}
\DeclarePunctuationPairs{period}{}
\DeclarePunctuationPairs{exclam}{*}
\DeclarePunctuationPairs{question}{*}
```

Da diese Funktion sprachspezifisch ist, muss `\DeclarePunctuationPairs` in dem Argument von `\DefineBibliographyExtras` (falls es in der Präambel verwendet wird) oder `\DeclareBibliographyExtras` (falls es in einem lokalen Modul verwendet wird) verwendet werden. Betrachten Sie §§ 3.7 and 4.9 für weitere Details. Beachten Sie, dass einige lokale Module Einstellungen verwenden könnten, welche sich von der Standardeinstellung des Paketes unterscheiden.<sup>32</sup>

`\DeclareQuotePunctuation{⟨characters⟩}`

Dieser Befehl kontrolliert die ‚American-style‘-Zeichensetzung. Die Verpackung von `\mkbibquote` von § 4.10.4 kann mit den Möglichkeiten der Zeichensetzung interagieren, wie sie in §§ 4.7.1, 4.7.3, 4.7.4 beschrieben werden. Satzzeichen hinter `\mkbibquote` werden mit in das Zitat hineingezogen, wenn sie mit `\DeclareQuotePunctuation` registriert wurden. Beachten Sie, dass `⟨characters⟩` keine unbegrenzte Liste von Zeichen darstellt. Gültige `⟨characters⟩` sind Punkt, Komma, Semikolon, Doppelpunkt, Ausrufezeichen und Fragezeichen. Hier ein Beispiel:

```
\DeclareQuotePunctuation{.,}
```

Den Befehl `\DeclareQuotePunctuation{}` auszuführen, ist äquivalent dazu, diese Funktion zu deaktivieren. Dies ist die Standardeinstellung des Paketes. Da diese Funktion sprachspezifisch ist, muss dieser Befehl im Argument von `\DefineBibliographyExtras` (falls er in der Präambel verwendet wird) oder `\DeclareBibliographyExtras` (falls er in einem lokalen Modul verwendet wird) verwendet werden. Betrachten Sie §§ 3.7 and 4.9 und § 3.9.1 für weitere Details.

`\uspunctuation` Dies ist eine Kurzform, um die Befehle `\DeclareQuotePunctuation` und `\DeclarePunctuationPairs` der unteren Ebenen zu benutzen, um die ‚American-style‘-Zeichensetzung zu aktivieren. Betrachten Sie § 3.9.1 für weitere Details. Diese Abkürzung ist nur aus Zweckmäßigkeitsgründen angegeben. Die wirksamen Einstellungen werden von den Befehlen der unteren Ebenen vorgenommen.

<sup>32</sup>In diesem Text benutzt das `american`-Modul verschiedene Einstellungen für die ‚American-style‘-Zeichensetzung.

`\stdpunctuation` Macht die Einstellungen rückgängig, die von `\uspunctuation` vorgenommen wurden, und stellt die Standardzeichensetzung wieder her. Um die Standardzeichensetzung zur standardisierten Einstellung zu erklären, benötigen Sie nur diesen Befehl, um einen vorher ausgeführten `\uspunctuation`-Befehl zu überschreiben. Betrachten Sie § 3.9.1 für weitere Details.

#### 4.7.6. Korrektur des Zeichensetzungstrackers

Die Funktionen des Zeichensetzungstrackers und der automatischen Großschreibung sind unter normalen Umständen sehr zuverlässig. Allerdings gibt es immer Grenzfälle, die manuelle Eingriffe erfordern. Solche typischen Fälle sind lokalisierte Zeichenfolgen, die als erstes Wort einer Fußnote abgebildet werden (dies wird für gewöhnlich, was die Großschreibung betrifft, als der Beginn eines Paragraphen behandelt, allerdings ist TeX zu diesem Zeitpunkt nicht im vertikalen Modus) oder Zeichensetzung nach Punkten, die nicht wirklich am Ende eines Satzes stehen (zum Beispiel würde ein Befehl wie `\addperiod` nach einer Ellipse wie "[. .]" nichts tun, außer die Klammern für den Zeichensetzungstracker unsichtbar zu machen). Benutzen Sie in solchen Fällen die folgenden Befehle in den Stildateien des Literaturverzeichnisses und der Zitierungen, um den Anfang oder die Mitte eines Satzes zu markieren, falls dies erforderlich ist:

`\bibsentence` Dieser Befehl markiert den Anfang eines Satzes. Eine lokalisierte Zeichenkette wird nach so einem Befehl automatisch groß geschrieben und der Zeichensetzungstracker wird zurückgesetzt. Das bedeutet, dieser Befehl verbirgt alle vorangehenden Satzzeichen vor dem Zeichensetzungstracker und erzwingt die Großschreibung.

`\midsentence` Dieser Befehl markiert die Mitte eines Satzes. Eine lokalisierte Zeichenkette wird nach solch einem Befehl nicht groß geschrieben und der Zeichensetzungstracker wird zurückgesetzt. Das bedeutet, dieser Befehl verbirgt alle vorangehenden Satzzeichen vor dem Zeichensetzungstracker und unterdrückt die Großschreibung.

`\midsentence*` Die Stern-Variante von `\midsentence` unterscheidet sich von der regulären darin, dass eine vorangehende Abkürzung nicht vor dem Zeichensetzungstracker verborgen wird. Das bedeutet, dass jede Zeichenfolge nach `\midsentence*` so tun wird, als würde ein Punkt nach einer Abkürzung vorausgehen. Alle anderen Satzzeichen werden vor dem Zeichensetzungstracker verborgen und die Großschreibung wird unterdrückt.

#### 4.8. Lokalisierte Zeichenketten

Lokalisierte Zeichenketten sind Schlüsselbegriffe wie ‚edition‘ (Ausgabe) oder ‚volume‘ (Umfang), welche von den lokalisierten Modulen von `biblatex` automatisch

übersetzt werden. Betrachten Sie § 4.9 für einen Überblick und § 4.9.2 für eine Liste von allen Zeichenfolgen, die standardmäßig unterstützt werden. Die Befehle in diesem Abschnitt werden benutzt, um den lokalisierten Ausdruck abzdrukken.

`\bibstring[wrapper]{key}`

Bildet die lokalisierte Zeichenkette *key* (Schlüssel) da ab, wo *key* ein Bezeichner von kleingeschriebenen Buchstaben ist (siehe: § 4.9.2). Die Zeichenfolge wird, wie vorgeschrieben, zu Anfang groß geschrieben. Betrachten Sie § 4.7.5 für weitere Details. Abhängig von dem `abbreviate`- (Abkürzungs-)Paket aus § 3.1.2.1, druckt `\bibstring` die kurze oder die lange Version der Zeichenfolge ab. Wenn lokalisierte Zeichenketten verschachtelt sind, d. h. wenn `\bibstring` in einer anderen Zeichenfolge verwendet wird, wird er sich wie `\bibxstring` verhalten. Wenn das *wrapper*- (Hüll-)Argument angegeben wird, wird die Zeichenkette an den *wrapper* zur Formatierung übergeben. Dies ist für Schriftartenbefehle, wie `\emph`, gedacht.

`\biblstring[wrapper]{key}`

Wie `\bibstring`, allerdings wird immer die lange Zeichenkette abgedruckt und die `abbreviate`- (Abkürzungs-)Option ignoriert.

`\bibsstring[wrapper]{key}`

Wie `\bibstring`, allerdings wird immer die kurze Zeichenkette abgedruckt und die `abbreviate`- (Abkürzungs-)Option ignoriert.

`\bibcpstring[wrapper]{key}`

Wie `\bibstring`, aber der Ausdruck wird zu Anfang groß geschrieben.

`\bibcplstring[wrapper]{key}`

Wie `\biblstring`, aber der Ausdruck wird zu Anfang groß geschrieben.

`\bibcpsstring[wrapper]{key}`

Wie `\bibsstring`, aber der Ausdruck wird zu Anfang groß geschrieben.

`\bibucstring[wrapper]{key}`

Wie `\bibstring`, aber der ganze Ausdruck wird groß geschrieben.

`\bibuclstring[wrapper]{key}`

Wie `\biblstring`, aber der ganze Ausdruck wird groß geschrieben.

`\bibucsstring[wrapper]{key}`

Wie `\bibsstring`, aber der ganze Ausdruck wird groß geschrieben.

`\biblcstring[⟨wrapper⟩]{⟨key⟩}`

Wie `\bibstring`, aber der ganze Ausdruck wird kleingeschrieben.

`\biblclstring[⟨wrapper⟩]{⟨key⟩}`

Wie `\bibstring`, aber der ganze Ausdruck wird kleingeschrieben.

`\biblcsstring[⟨wrapper⟩]{⟨key⟩}`

Wie `\bibsstring`, aber der ganze Ausdruck wird kleingeschrieben.

`\bibxstring{⟨key⟩}`

Eine vereinfachte, aber erweiterbare Version von `\bibstring`. Beachten Sie, dass diese Variante die automatische Großschreibung des ersten Buchstaben nicht unterstützt, noch funktioniert sie im Zusammenhang mit dem Zeichensetzungstracker. Es ist für Spezialfälle gedacht, in welchen Zeichenfolgen verschachtelt sind oder eine erweiterte lokalisierte Zeichenkette für einen Test benötigt wird.

`\bibxlstring[⟨wrapper⟩]{⟨key⟩}`

Wie `\bibxstring`, aber es wird immer die lange Zeichenkette verwendet und die `abbreviate`- (Abkürzungs-)Option ignoriert.

`\bibxsstring[⟨wrapper⟩]{⟨key⟩}`

Wie `\bibxstring`, aber es wird immer die kurze Zeichenkette verwendet und die `abbreviate`- (Abkürzungs-)Option ignoriert.

#### 4.9. Lokalisierte Module

Ein lokalisiertes Modul bietet Übersetzungen für Schlüsselbegriffe wie ‚edition‘ oder ‚volume‘, genauso wie Definitionen für sprachspezifische Funktionen wie das Datumsformat und Ordinalzeichen. Diese Definitionen werden in Dateien zur Verfügung gestellt, die auf `lbx` enden. Der Basisname der Datei muss ein Name einer Sprache sein, der dem `babel`-Paket bekannt ist. Die `lbx`-Dateien können auch dazu verwendet werden, um die Namen der Sprachen von `babel` den "backendModulen des `biblatex`-Pakets zuzuordnen. Alle lokalisierten Module werden auf Anfrage des Dokumentrumpfes geladen. Beachten Sie, dass die Inhalte der Datei in einer Gruppe verarbeitet werden und dass der Code der Kategorie des Zeichens `@` vorübergehend auf ‚letter‘ eingestellt wird.

### 4.9.1. Lokalisierte Befehle

Die Versionen der Benutzerebene der lokalisierten Befehle wurden bereits in § 3.7 angesprochen. Wenn die lokalisierten Befehle in l<sub>bx</sub>-Dateien verwendet werden, unterscheidet sich deren Syntax von der gewöhnlichen Syntax in der Präambel und der von der Konfigurationsdatei. Wenn sie in lokalisierten Dateien verwendet werden, muss man die *⟨language⟩* (Sprache) nicht angeben, da den Wörtern bereits durch den Namen der l<sub>bx</sub> eine Sprache zugeordnet wurde.

```
\DeclareBibliographyStrings{⟨definitions⟩}
```

Dieser Befehl ist nur in l<sub>bx</sub>-Dateien verfügbar. Er wird verwendet, um lokalisierte Zeichenketten zu definieren. Die *⟨definitions⟩* (Definitionen) bestehen aus *⟨key⟩=⟨value⟩*-Paaren, welche einen Ausdruck einem Bezeichner zuordnen. Eine vollständige Liste aller Schlüssel, die von der Standardeinstellung unterstützt werden, steht in § 4.9.2. Beachten Sie, dass die Syntax des „Wertes“ sich von den l<sub>bx</sub>-Dateien unterscheidet. Der einem Schlüssel zugewiesene „Wert“ besteht aus zwei Ausdrücken, jeder von ihnen steht in einem zusätzlichen Paar von Klammern. Am besten wird dies am Beispiel gezeigt:

```
\DeclareBibliographyStrings{%
bibliography = {{Bibliography}{Bibliography}},
shorthands = {{List of Abbreviations}{Abbreviations}},
editor = {{editor}{ed.}},
editors = {{editors}{eds.}}, }
```

Der erste Wert ist der lange, ausgeschriebene Ausdruck, der zweite ist eine Abkürzung oder Kurzform. Beide Wörter müssen immer gegeben sein, selbst wenn sie identisch sein sollten, falls ein Ausdruck niemals (oder immer) abgekürzt wird. Abhängig von den Einstellungen der Optionen des `abbreviate`- (Abkürzungs-)Pakets (siehe § 3.1.2.1), wählt `biblatex` einen Ausdruck aus, wenn die l<sub>bx</sub>-Datei geladen wird. Außerdem gibt es einen speziellen Schlüssel namens `inherit` (Vererbung), welcher die Zeichenketten aus einer anderen Sprache kopiert. Dies ist für Sprachen gedacht, die sich nur in wenigen Ausdrücken unterscheiden, wie zum Beispiel die deutsche und die österreichische Sprache oder amerikanisches und britisches Englisch. Hier zum Beispiel die komplette Definition für österreichisches Deutsch:

```
\DeclareBibliographyStrings{%
inherit = {german},
january = {{J\"anner}{J\"an.}}, }
```

Die obigen Beispiele sind etwas vereinfacht. Echte lokalisierte Dateien sollten die Zeichensetzungs- und Formatierungsbefehle benutzen, die in §§ 4.7.3 and 3.8 erklärt werden, anstatt wörtliche Zeichensetzung zu verwenden. Hier ein Auszug aus einer echten Lokalisierungsdatei:

```

bibliography = {{Bibliography}{Bibliography}},
shorthands = {{List of Abbreviations}{Abbreviations}},
editor = {{editor}{ed\addot}},
editors = {{editors}{eds\addot}},
byeditor = {{edited by}{ed\addotsspace by}},
mathesis = {{Master's thesis}{MA\addabbrvspace
thesis}},

```

Beachten Sie, wie mit den Abkürzungspunkten, der Leerzeichensetzung in abgekürzten Ausdrücken und der Großschreibung im obigen Beispiel umgegangen wird. Die Großschreibung des ersten Buchstaben eines Wortes sollte sich nicht verändern, wenn dieses in der Mitte eines Satzes geschrieben wird. Das `biblatex`-Paket wird das erste Wort automatisch groß schreiben, wenn es zu Beginn eines Satzes steht. Betrachten Sie `\DeclareCapitalPunctuation` in § 4.7.5 für weitere Details. Ausdrücke, die für den Gebrauch in Überschriften gedacht sind, werden gesondert behandelt. Diese sollten in einer Art und Weise groß geschrieben werden, die für die Titelgebung geeignet ist, und sollten nicht abgekürzt werden (trotzdem können sie eine Kurzform haben).

```
\InheritBibliographyStrings{⟨language⟩}
```

Dieser Befehl ist nur in `lbx`-Dateien verfügbar. Er kopiert die lokalisierten Zeichenfolgen für `⟨language⟩` in die gegenwärtige Sprache, wie sie vom Namen der `lbx`-Datei vorgegeben wird.

```
\DeclareBibliographyExtras{⟨code⟩}
```

Dieser Befehl ist nur in `lbx`-Dateien verfügbar. Er wird benutzt, um sprachspezifische Funktionen wie das Datumsformat und Ordinalzahlen anzupassen. Der `⟨code⟩`, welcher ein willkürlicher LaTeX-Kode sein kann, besteht für gewöhnlich aus Neudefinitionen der Formatierungsbefehle von § 4.10.2.

```
\UndeclareBibliographyExtras{⟨code⟩}
```

Dieser Befehl ist nur in `lbx`-Dateien verfügbar. Er wird benutzt, um jegliche Formatierungsbefehle wiederherzustellen, die mit `\DeclareBibliographyExtras` modifiziert wurden. Wenn ein neu definierter Befehl in § 4.10.2 eingeschlossen ist, muss man ihn nicht wieder in seinen Ausgangszustand zurückversetzen, da er bereits in allen Sprachmodulen enthalten ist.

```
\InheritBibliographyExtras{⟨language⟩}
```

Dieser Befehl ist nur in `lbx`-Dateien verfügbar. Er kopiert die Extras der `⟨language⟩` (Sprache) des Literaturverzeichnisses in die aktuelle Sprache, die vom Namen der `lbx`-Datei vorgegeben wird.

`\DeclareHyphenationExceptions{⟨text⟩}`

Dieser Befehl entspricht § 3.7. Der Unterschied liegt darin, dass er nur in l<sub>bx</sub>-Dateien verfügbar ist und dass das *⟨language⟩* Argument weggelassen wird. Die Ausnahmen der Silbentrennung werden die Sprache der l<sub>bx</sub>-Datei beeinflussen, die aktuell verwendet wird.

`\DeclareRedundantLanguages{⟨language, language, ...⟩}{⟨babel, babel, ...⟩}`

Dieser Befehl stellt die Sprachzuordnungen zur Verfügung, die von der `clearlang`-Option von § 3.1.2.1 benötigt werden. Die *⟨language⟩* (Sprache) ist die Zeichenfolge, die im `language`-Feld steht (ohne den optionalen `lang`-Präfix); *⟨babel⟩* bezeichnet hierbei die von `babel` verwendete Sprache, die auch im optionalen Argument von `\usepackage` steht, wenn `babel` geladen wird. Dieser Befehl kann entweder in l<sub>bx</sub>-Dateien oder in der Präambel des Dokumentes verwendet werden. Hier einige Beispiele:

```
\DeclareRedundantLanguages{french}{french}
\DeclareRedundantLanguages{german}{german,ngerman,austrian,naustrian}
\DeclareRedundantLanguages{english,american}{english,american,british,
canadian,australian,newzealand,USenglish,UKenglish}
```

Beachten Sie, dass diese Funktion global von der `clearlang`-Option aus § 3.1.2.1 aktiviert werden muss. Wenn sie deaktiviert ist, werden alle Zuordnungen ignoriert. Wenn die Parameter von *⟨babel⟩* leer sind, wird `biblatex` die Zuordnungen der jeweiligen *⟨language⟩* (Sprache) löschen. Das bedeutet, dass diese Funktion nur für diese Sprache deaktiviert ist.

`\DeclareLanguageMapping{⟨language⟩}{⟨file⟩}`

Dieser Befehl ordnet einen `babel`-Sprachbezeichner einer l<sub>bx</sub>-Datei zu. Die *⟨language⟩* (Sprache) muss der Name einer Sprache sein, die dem `babel`-Paket bekannt ist, also einer der Bezeichner aufgelistet in Tabelle 1. Das *⟨file⟩*-Argument ist der Name einer alternativen l<sub>bx</sub>-Datei ohne die Endung `.lbx`. Es ist möglich, eine Zuordnung mehr als einmal zu deklarieren. Nachfolgende Deklarierungen überschreiben vorhergehende einfach. Dieser Befehl kann nur in der Präambel verwendet werden. Betrachten Sie § 4.11.8 für weitere Details.

`\NewBibliographyString{⟨key⟩}`

Dieser Befehl, der sowohl in der Präambel (inklusive `cbx`- und `bbx`-Dateien) als auch in l<sub>bx</sub>-Dateien verwendet werden kann, deklariert neue lokalisierte Zeichenketten, das heißt, er initialisiert einen neuen *⟨key⟩* (Schlüssel), der in den *⟨definitions⟩* (Definitionen) von `\DefineBibliographyStrings` verwendet wird. Das *⟨key⟩*- (Schlüssel-)Argument kann auch eine mit Kommas getrennte Auflistung der Schlüsselnamen sein. Die in § 4.9.2 aufgeführten Schlüssel sind standardmäßig definiert.

## 4.9.2. Lokalisierte Schlüssel

Die lokalisierten Schlüssel in diesem Abschnitt sind standardmäßig definiert und werden von den lokalisierten Dateien überdeckt, die von `biblatex` kommen. Beachten Sie, dass diese Zeichenketten nur bei Zitaten, dem Literaturverzeichnis und der Liste von Kurzformen verfügbar sind. Alle Ausdrücke sollten wie gewohnt zu Beginn groß geschrieben werden. `biblatex` schreibt sie automatisch groß, falls sie am Anfang eines Satzes stehen. Die einzige Ausnahme bilden die drei Wörter, die für die Verwendung in Überschriften gedacht sind.

**4.9.2.1. Überschriften** Die folgenden Zeichenketten sind besonders, da sie für die Verwendung in Überschriften gedacht sind und via Makro global zugänglich gemacht sind. Aus diesem Grund sollten sie groß geschrieben werden, wenn sie in Überschriften stehen und dürfen keine lokalen Befehle enthalten, die Teil des Interfaces des Autors von `biblatex` sind.

- `bibliography` Der Begriff ‚bibliography‘ (Literaturverzeichnis), auch verfügbar als `\bibname`.
- `references` Der Begriff ‚references‘ (Verweise), auch verfügbar als `\refname`.
- `shorthands` Der Begriff ‚list of shorthands‘ oder ‚list of abbreviations‘ (Liste der Abkürzungen), auch verfügbar als `\losname`.

**4.9.2.2. Aufgaben, ausgedrückt als Funktionen** Die folgenden Schlüssel beziehen sich auf Aufgaben, die als Funktion (‚editor‘ (Herausgeber), ‚translator‘ (Übersetzer)) und nicht als eine Aktion (‚edited by‘ (herausgegeben von), ‚translated by‘ (übersetzt von)) ausgedrückt werden.

- `editor` Der Ausdruck ‚editor‘ (Herausgeber) bezieht sich auf den Haupteditor. Dies ist die allgemeinste Aufgabe des Editors.
- `editors` Die Pluralform von `editor`.
- `compiler` Der Ausdruck ‚compiler‘ (Kompilierer) bezieht sich auf einen Editor, dessen Aufgabe darin besteht, eine Arbeit zu kompilieren.
- `compilers` Die Plural-Form von `compiler`.
- `founder` Der Ausdruck ‚founder‘ (Gründer) bezieht sich auf einen „gründenden“ Editor.
- `founders` Die Pluralform von `founder`.
- `continuator` Ein Ausdruck wie ‚continuator‘ (Fortsetzer), ‚continuation‘ (Fortsetzung) oder ‚continued‘ (fortgesetzt) bezieht sich auf einen früheren Editor, der die Arbeit eines „gründenden“ Editors fortsetzt, aber vom derzeitigen Editor ersetzt wird.
- `continuators` Die Pluralform von `continuator`.
- `redactor` Der Ausdruck ‚redactor‘ (Redakteur) bezieht sich auf einen Sekundäreditor.

|                      |                                                                                                                                                                                   |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>redactors</b>     | Die Plural-Form von redactor.                                                                                                                                                     |
| <b>collaborator</b>  | Ein Ausdruck wie ‚collaborator‘ (Mitarbeiter), ‚collaboration‘ (Zusammenarbeit), ‚cooperator‘ (Kooperator) oder ‚cooperation‘ (Kooperation) bezieht sich auf einen Sekundäreditor |
| <b>collaborators</b> | Die Pluralform von collaborator.                                                                                                                                                  |
| <b>translator</b>    | Der Ausdruck ‚translator‘ (Übersetzer).                                                                                                                                           |
| <b>translators</b>   | Die Pluralform von translator.                                                                                                                                                    |
| <b>commentator</b>   | Der Ausdruck ‚commentator‘ (Kommentator) bezieht sich auf den Autor eines Kommentars zu einer Arbeit.                                                                             |
| <b>commentators</b>  | Die Pluralform von commentators.                                                                                                                                                  |
| <b>annotator</b>     | Der Ausdruck ‚annotator‘ (Kommentator) bezieht sich auf den Autor einer Anmerkung zu einer Arbeit.                                                                                |
| <b>annotators</b>    | Die Pluralform von annotators.                                                                                                                                                    |

**4.9.2.3. Verkettete Aufgaben des Editors, ausgedrückt als Funktionen** Die folgenden Schlüssel ähneln editor, translator etc. in ihrer Funktion. Sie dienen dazu, weitere Aufgaben des Editors anzugeben, wie zum Beispiel ‚editor and translator‘ (Herausgeber und Übersetzer) und ‚editor and foreword‘ (Herausgeber und Vorwort).

|                  |                                                        |
|------------------|--------------------------------------------------------|
| <b>editortr</b>  | Wird benutzt, wenn editor/translator identisch sind.   |
| <b>editorstr</b> | Die Pluralform von editortr.                           |
| <b>editorco</b>  | Wird benutzt, wenn editor/commentator identisch sind.  |
| <b>editorsco</b> | Die Pluralform von editorco.                           |
| <b>editoran</b>  | Wird benutzt, wenn editor/annotator identisch sind.    |
| <b>editorsan</b> | Die Pluralform von editoran.                           |
| <b>editorin</b>  | Wird benutzt, wenn editor/introduction identisch sind. |
| <b>editorsin</b> | Die Pluralform von editorin.                           |
| <b>editorfo</b>  | Wird benutzt, wenn editor/foreword identisch sind.     |
| <b>editorsfo</b> | Die Pluralform von editorfo.                           |
| <b>editoraf</b>  | Wird benutzt, wenn editor/aftword identisch sind.      |
| <b>editorsaf</b> | Die Pluralform von editoraf.                           |

Schlüssel für die Kombination von editor/translator/⟨role⟩:

|                   |                                                                  |
|-------------------|------------------------------------------------------------------|
| <b>editortrco</b> | Wird benutzt, wenn editor/translator/commentator identisch sind. |
|-------------------|------------------------------------------------------------------|

- editorstrco** Die Pluralform von editortrco.
- editortran** Wird benutzt, wenn editor/translator/annotator identisch sind.
- editorstran** Die Pluralform von editortran.
- editortrin** Wird benutzt, wenn editor/translator/introduction identisch sind.
- editorstrin** Die Pluralform von editortrin.
- editortrfo** Wird benutzt, wenn editor/translator/foreword identisch sind.
- editorstrfo** Die Pluralform von editortrfo.
- editortraf** Wird benutzt, wenn editor/translator/aftword identisch sind.
- editorstraf** Die Pluralform von editortraf.

Schlüssel für die Kombination von editor/commentator/⟨role⟩:

- editorcoin** Wird benutzt, wenn editor/commentator/introduction identisch sind.
- editorscoin** Die Pluralform von editorcoin.
- editorcofo** Wird benutzt, wenn editor/commentator/foreword identisch sind.
- editorscofo** Die Pluralform von editorcofo.
- editorcoaf** Wird benutzt, wenn editor/commentator/aftword identisch sind.
- editorscoaf** Die Pluralform von editorcoaf.

Schlüssel für die Kombination von editor/annotator/⟨role⟩:

- editoranin** Wird benutzt, wenn editor/annotator/introduction identisch sind.
- editorsanin** Die Pluralform von editoranin.
- editoranfo** Wird benutzt, wenn editor/annotator/foreword identisch sind.
- editorsanfo** Die Pluralform von editoranfo.
- editoranaf** Wird benutzt, wenn editor/annotator/aftword identisch sind.
- editorsanaf** Die Pluralform von editoranaf.

Schlüssel für die Kombination von editor/translator/commentator/⟨role⟩:

- editortrcoin** Wird benutzt, wenn editor/translator/commentator/introduction identisch sind.
- editorstrcoin** Die Pluralform von editortrcoin.
- editortrcofo** Wird benutzt, wenn editor/translator/commentator/foreword identisch sind.
- editorstrcofo** Die Pluralform von editortrcofo.
- editortrcoaf** Wird benutzt, wenn editor/translator/commentator/aftword identisch sind.

**editorstrcoaf** Die Pluralform von editortrcoaf.

Schlüssel für die Kombination von editor/annotator/commentator/⟨role⟩:

**editortranin** Wird benutzt, wenn editor/annotator/commentator/introduction identisch sind.

**editorstranin** Die Pluralform von editortranin.

**editortranfo** Wird benutzt, wenn editor/annotator/commentator/foreword identisch sind.

**editorstranfo** Die Pluralform von editortranfo.

**editortranaf** Wird benutzt, wenn editor/annotator/commentator/aftword identisch sind.

**editorstranaf** die Pluralform von editortranaf.

#### 4.9.2.4. Verkettete Aufgaben für den Übersetzer, ausgedrückt als Funktionen

Die folgenden Schlüssel ähneln translator (Übersetzer) in ihrer Funktion. Sie dienen dazu, zusätzliche Aufgaben des Übersetzers anzugeben wie zum Beispiel ‚translator and commentator‘ (Übersetzer und Kommentator) und ‚translator and introduction‘ (Übersetzer und Einführung).

**translatorco** Wird benutzt, wenn translator/commentator identisch sind.

**translatorsco** Die Pluralform von translatorco.

**translatorsan** Wird benutzt, wenn translator/annotator identisch sind.

**translatorsan** Die Pluralform von translatorsan.

**translatorsin** Wird benutzt, wenn translator/introduction identisch sind.

**translatorsin** Die Pluralform von translatorsin.

**translatorfo** Wird benutzt, wenn translator/foreword identisch sind.

**translatorsfo** Die Pluralform von translatorfo.

**translatorsaf** Wird benutzt, wenn translator/aftword identisch sind.

**translatorsaf** Die Pluralform von translatorsaf.

Schlüssel für die Kombination von translator/commentator/⟨role⟩:

**translatorcoin** Wird benutzt, wenn translator/commentator/introduction identisch sind.

**translatorscoin** Die Pluralform von translatorcoin.

**translatorcofo** Wird benutzt, wenn translator/commentator/foreword identisch sind.

**translatorscofo** Die Pluralform von translatorcofo.

**translatorcoaf** Wird benutzt translator/commentator/aftword identisch sind.

|                              |                                                                                   |
|------------------------------|-----------------------------------------------------------------------------------|
| <code>translatorscoaf</code> | Die Pluralform von <code>translatorcoaf</code> .                                  |
|                              | Schlüssel für die Kombination von <code>translator/annotator/⟨role⟩</code> :      |
| <code>translatoranin</code>  | Wird benutzt, wenn <code>translator/annotator/introduction</code> identisch sind. |
| <code>translatorsanin</code> | Die Pluralform von <code>translatoranin</code> .                                  |
| <code>translatoranfo</code>  | Wird benutzt, wenn <code>translator/annotator/foreword</code> identisch sind.     |
| <code>translatorsanfo</code> | Die Pluralform von <code>translatoranfo</code> .                                  |
| <code>translatoranaf</code>  | Wird benutzt, wenn <code>translator/annotator/aftword</code> identisch sind.      |
| <code>translatorsanaf</code> | Die Pluralform von <code>translatoranaf</code> .                                  |

**4.9.2.5. Aufgaben, ausgedrückt als Aktionen** Die folgenden Schlüssel beziehen sich auf Aufgaben, die eher als Aktionen (`,edited by'` (herausgegeben von), `,translated by'` (übersetzt von)) als Funktionen (`,editor'` (Herausgeber), `,translator'` (Übersetzer)) ausgedrückt werden.

|                             |                                                                                                                                    |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <code>byauthor</code>       | Der Ausdruck <code>,erstellt] von ⟨name⟩'</code> .                                                                                 |
| <code>byeditor</code>       | Der Ausdruck <code>,bearbeitet von ⟨name⟩'</code> .                                                                                |
| <code>bycompiler</code>     | Der Ausdruck <code>,kompiliert von ⟨name⟩'</code> .                                                                                |
| <code>byfounder</code>      | Der Ausdruck <code>,gegründet von ⟨name⟩'</code> .                                                                                 |
| <code>bycontinuator</code>  | Der Ausdruck <code>,fortgesetzt von ⟨name⟩'</code> .                                                                               |
| <code>byredactor</code>     | Der Ausdruck <code>,redigiert ⟨name⟩'</code> .                                                                                     |
| <code>bycollaborator</code> | Ein Ausdruck wie <code>,in Zusammenarbeit mit ⟨name⟩ entstanden'</code> oder <code>,in Kooperation mit ⟨name⟩ entstanden'</code> . |
| <code>bytranslator</code>   | Der Ausdruck <code>,übersetzt von ⟨name⟩'</code> oder <code>,aus dem ⟨language⟩ übersetzt von⟨name⟩'</code> .                      |
| <code>bycommentator</code>  | Der Ausdruck <code>,kommentiert von ⟨name⟩'</code> .                                                                               |
| <code>byannotator</code>    | Der Ausdruck <code>,annotiert von ⟨name⟩'</code> .                                                                                 |

**4.9.2.6. Verkettete Editoraufgaben, ausgedrückt als Aktionen** Die folgenden Schlüssel ähneln in ihrer Funktion `byeditor`, `bytranslator` etc. Sie werden benutzt, um zusätzliche Aufgaben des Editors anzugeben, so zum Beispiel `,edited and translated by'` (herausgegeben und übersetzt von), `,edited and furnished with an introduction by'` (herausgegeben und mit einer Einleitung versehen von), `,edited, with a foreword, by'` (mit einem Vorwort versehen und herausgegeben von).

|                         |                                                                   |
|-------------------------|-------------------------------------------------------------------|
| <code>byeditortr</code> | Wird benutzt, wenn <code>editor/translator</code> identisch sind. |
|-------------------------|-------------------------------------------------------------------|

- byeditorco Wird benutzt, wenn editor/commentator identisch sind.
- byeditoran Wird benutzt, wenn editor/annotator identisch sind.
- byeditorin Wird benutzt, wenn editor/introduction identisch sind.
- byeditorfo Wird benutzt, wenn editor/foreword identisch sind.
- byeditoraf Wird benutzt, wenn editor/aftword identisch sind.

Schlüssel für die Kombination von editor/translator/⟨role⟩:

- byeditortrco Wird benutzt, wenn editor/translator/commentator identisch sind.
- byeditortran Wird benutzt, wenn editor/translator/annotator identisch sind.
- byeditortrin Wird benutzt, wenn editor/translator/introduction identisch sind.
- byeditortrfo Wird benutzt, wenn editor/translator/foreword identisch sind.
- byeditortraf Wird benutzt, wenn editor/translator/aftword identisch sind.

Schlüssel für die Kombination von editor/commentator/⟨role⟩:

- byeditorcoin Wird benutzt, wenn editor/commentator/introduction identisch sind.
- byeditorcofo Wird benutzt, wenn editor/commentator/foreword identisch sind.
- byeditorcoaf Wird benutzt, wenn editor/commentator/aftword identisch sind.

Schlüssel für die Kombination von editor/annotator/⟨role⟩:

- byeditoranin Wird benutzt, wenn editor/annotator/introduction identisch sind.
- byeditoranfo Wird benutzt, wenn editor/annotator/foreword identisch sind.
- byeditoranaf Wird benutzt, wenn editor/annotator/aftword identisch sind.

Schlüssel für die Kombination von editor/translator/commentator/⟨role⟩:

- byeditortrcoin Wird benutzt, wenn editor/translator/commentator/introduction identisch sind.
- byeditortrcofo Wird benutzt, wenn editor/translator/commentator/foreword identisch sind.
- byeditortrcoaf Wird benutzt, wenn editor/translator/commentator/aftword identisch sind.

Schlüssel für die Kombination von editor/translator/annotator/⟨role⟩:

- byeditortranin Wird benutzt, wenn editor/annotator/commentator/introduction identisch sind.
- byeditortranfo Wird benutzt, wenn editor/annotator/commentator/foreword identisch sind.
- byeditortranaf Wird benutzt, wenn editor/annotator/commentator/aftword identisch sind.

**4.9.2.7. Verkettete Aufgaben des Übersetzers, ausgedrückt als Aktionen** Die folgenden Schlüssel ähneln `bytranslator` in ihrer Funktion. Sie werden verwendet, um zusätzliche Aufgaben des Übersetzers anzugeben, so zum Beispiel ‚translated and commented by‘ (übersetzt und kommentiert von), ‚translated and furnished with an introduction by‘ (übersetzt und mit einer Einleitung versehen von), ‚translated, with a foreword, by‘ (mit einem Vorwort versehen und übersetzt von).

- `bytranslatorco` Wird benutzt, wenn `translator/commentator` identisch sind.
- `bytranslatoran` Wird benutzt, wenn `translator/annotator` identisch sind.
- `bytranslatorin` Wird benutzt, wenn `translator/introduction` identisch sind.
- `bytranslatorfo` Wird benutzt, wenn `translator/foreword` identisch sind.
- `bytranslatoraf` Wird benutzt, wenn `translator/aftword` identisch sind.

Schlüssel für die Kombination von `translator/commentator/⟨role⟩`:

- `bytranslatorcoin` Wird benutzt, wenn `translator/commentator/introduction` identisch sind.
- `bytranslatorcofo` Wird benutzt, wenn `translator/commentator/foreword` identisch sind.
- `bytranslatorcoaf` Wird benutzt, wenn `translator/commentator/aftword` identisch sind.

Schlüssel für die Kombination von `translator/annotator/⟨role⟩`:

- `bytranslatoranin` Wird benutzt, wenn `translator/annotator/introduction` identisch sind.
- `bytranslatoranfo` Wird benutzt, wenn `translator/annotator/foreword` identisch sind.
- `bytranslatoranaf` Wird benutzt, wenn `translator/annotator/aftword` identisch sind.

**4.9.2.8. Aufgaben, ausgedrückt als Objekte** Aufgaben, die mit ergänzendem Material in Verbindung stehen, können eher als Objekte (‚with a commentary by‘ (mit einem Kommentar von)) als Funktionen (‚commentator‘ (Kommentator)) oder Aktionen (‚commented by‘ (kommentiert von)) ausgedrückt werden.

- `withcommentator` Der Ausdruck ‚mit einem Kommentar von `⟨name⟩`‘.
- `withannotator` Der Ausdruck ‚mit Anmerkungen von `⟨name⟩`‘.
- `withintroduction` Der Ausdruck ‚mit einer Einleitung von `⟨name⟩`‘.
- `withforeword` Der Ausdruck ‚mit einem Vorwort von `⟨name⟩`‘.
- `withafterword` Der Ausdruck ‚mit einem Nachwort von `⟨name⟩`‘.

#### 4.9.2.9. Ergänzendes Material

|              |                            |
|--------------|----------------------------|
| commentary   | Der Ausdruck ‚Kommentar‘.  |
| annotations  | Der Ausdruck ‚Anmerkung‘.  |
| introduction | Der Ausdruck ‚Einleitung‘. |
| foreword     | Der Ausdruck ‚Vorwort‘.    |
| afterword    | Der Ausdruck ‚Nachwort‘.   |

#### 4.9.2.10. Veröffentlichungsdetails

|           |                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| volume    | Der Ausdruck ‚volume‘ (Umfang) auf ein Buch bezogen.                                                                                                                                                |
| volumes   | Die Pluralform von volume.                                                                                                                                                                          |
| jourvol   | Der Ausdruck ‚volume‘ (Umfang) bezogen auf eine Zeitschrift.                                                                                                                                        |
| jourser   | Der Ausdruck ‚series‘ (Serie) bezogen auf eine Zeitschrift.                                                                                                                                         |
| newseries | Der Ausdruck ‚new series‘ (neue Serie) bezogen auf eine Fachzeitschrift.                                                                                                                            |
| oldseries | Der Ausdruck ‚old series‘ (alte Serie) bezogen auf eine Zeitschrift.                                                                                                                                |
| edition   | Der Ausdruck ‚edition‘ (Ausgabe).                                                                                                                                                                   |
| in        | Der Ausdruck ‚in‘, bezogen auf den Titel einer Arbeit, die als Teil einer anderen Arbeit veröffentlicht wurde, zum Beispiel ‚ <i>&lt;title of article&gt;</i> in <i>&lt;title of journal&gt;</i> ‘. |
| inseries  | Der Ausdruck ‚in‘, wie er in Ausdrücken wie ‚volume <i>&lt;number&gt;</i> in <i>&lt;name of series&gt;</i> ‘ verwendet wird.                                                                        |
| ofseries  | Der Ausdruck ‚of‘, wie er in Ausdrücken wie ‚volume <i>&lt;number&gt;</i> of <i>&lt;name of series&gt;</i> ‘ verwendet wird.                                                                        |
| number    | Der Ausdruck ‚number‘ (Nummer), bezogen auf die Ausgabe einer Zeitschrift.                                                                                                                          |
| chapter   | Der Ausdruck ‚chapter‘ (Kapitel) bezogen auf das Kapitel in einem Buch.                                                                                                                             |
| version   | Der Ausdruck ‚version‘ (Version) bezogen auf eine Versionsnummer.                                                                                                                                   |
| reprint   | Der Ausdruck ‚reprint‘ (Neudruck).                                                                                                                                                                  |
| reprintof | Der Ausdruck ‚reprint of <i>&lt;title&gt;</i> ‘.                                                                                                                                                    |
| reprintas | Der Ausdruck ‚reprinted as <i>&lt;title&gt;</i> ‘.                                                                                                                                                  |

#### 4.9.2.11. Veröffentlichungszustand

|               |                                                                                                                                                   |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| inpress       | Der Ausdruck ‚in press‘ (im Druck).                                                                                                               |
| inpreparation | Der Ausdruck ‚in preparation‘ (in der Vorbereitung) bezogen auf ein Manuskript, welches auf die Veröffentlichung vorbereitet wird.                |
| submitted     | Der Ausdruck ‚submitted‘ (eingereicht) bezogen auf einen Artikel oder eine Arbeit, die bei einer Zeitschrift oder einer Tagung eingereicht wurde. |

#### 4.9.2.12. Pagination

- page** Der Ausdruck ‚page‘ (Seite).
- pages** Die Pluralform von **page**.
- column** Der Ausdruck ‚column‘ (Rubrik) bezogen auf eine Rubrik auf einer Seite.
- columns** Die Pluralform von **column**.
- section** Der Ausdruck ‚section‘ (Abschnitt) bezogen auf die Unterteilung eines Dokuments (üblicherweise abgekürzt mit §).
- sections** Die Plural-Form von **section** (üblicherweise abgekürzt mit §§).
- paragraph** Der Ausdruck ‚paragraph‘ (Absatz) (das bedeutet ein Textblock, nicht zu verwechseln mit **section**).
- paragraphs** Die Pluralform von **paragraph**.
- verse** Der Ausdruck ‚verse‘ (Strophe), wie er benutzt wird, wenn das Werk mit Strophennummern zitiert wird.
- verses** Die Plural-Form von **verse**.
- line** Der Ausdruck ‚line‘ (Linie), wie er benutzt wird, wenn eine Arbeit nach Liniennummern zitiert wird.
- lines** Die Pluralform von **line**.

**4.9.2.13. Arten** Die folgenden Schlüssel werden gewöhnlicherweise im `type` Eingabefeld von `@thesis`, `@report`, `@misc` und anderen Einträgen verwendet:

- mathesis** Ein äquivalenter Ausdruck zu dem Ausdruck ‚Master’s thesis‘ (Masterarbeit).
- phdthesis** Der Ausdruck ‚PhD thesis‘, ‚PhD dissertation‘, ‚doctoral thesis‘ etc. (alle gleichbedeutend mit Doktorarbeit).
- techreport** Der Ausdruck ‚technical report‘ (technischer Bericht).
- resreport** Der Ausdruck ‚research report‘ (Forschungsbericht).
- software** Der Ausdruck ‚computer software‘ (Computersoftware).
- datacd** Der Ausdruck ‚data CD‘ or ‚CD-ROM‘ (Daten-CD oder CD-Rom).
- audiocd** Der Ausdruck ‚audio CD‘ (Musik-CD).

#### 4.9.2.14. Verschiedenes

- and** Der Begriff ‚and‘ (und), wie er zum Beispiel in einer Liste von Autoren und Herausgebern verwendet wird.
- andothers** Der Ausdruck ‚and others‘ oder ‚et alii‘ (und weitere) wird verwendet, um die Verkürzung einer Namensliste zu markieren.

**andmore** Wie *andothers*, wird allerdings verwendet, um die Verkürzung einer Literaturliste zu markieren.

**4.9.2.15. Beschriftungen** Die folgenden Zeichenketten sind für die Verwendung in Beschriftungen wie zum Beispiel ‚Address: *<url>*‘ (url-Adresse) oder ‚Abstract: *<abstract>*‘ (Zusammenfassung) vorgesehen:

- url** Der Begriff ‚address‘ im Sinne einer Internetadresse.
- urlseen** Ein Ausdruck wie ‚accessed on *<date>*‘, ‚abgerufen am *<date>*‘, ‚besucht am *<date>*‘, der sich auf das Aufrufdatum einer Online-Ressource bezieht.
- file** Der Begriff ‚file‘ (Datei).
- library** Der Begriff ‚library‘ (Bibliothek).
- abstract** Der Begriff ‚abstract‘ (Zusammenfassung/Auszug).
- annotation** Der Begriff ‚annotations‘ (Anmerkungen).

**4.9.2.16. Zitierungen** Traditionelle wissenschaftliche Ausdrücke, die in Zitaten verwendet werden:

- idem** Äquivalent zum lateinischen Ausdruck ‚idem‘ (‚the same [person]‘ (dieselbe Person)).
- idemsf** Die weibliche Singularform von *idem*.
- idemsm** Die männliche Singularform von *idem*.
- idemsn** Die neutrale Singularform *idem*.
- idempf** Die weibliche Pluralform von *idem*.
- idempm** Die männliche Pluralform von *idem*.
- idempn** Die neutrale Pluralform von *idem*.
- idempp** Die Pluralform von *idem*, geeignet für eine Liste von Namen mit unterschiedlichen Geschlechtern.
- ibidem** Äquivalent zum lateinischen Ausdruck ‚ibidem‘ (‚in the same place‘ (am selben Ort)).
- opcit** Äquivalent zum lateinischen Ausdruck ‚opere citato‘ (‚[in] the work [already] cited‘ (im Werk bereits zitiert)).
- loccit** Äquivalent zum lateinischen Ausdruck ‚loco citato‘ (‚[at] the place [already] cited‘ (an der Stelle bereits zitiert)).
- confer** Äquivalent zum lateinischen Ausdruck ‚confer‘ (‚compare‘ (vergleichen)).

- sequens** Äquivalent zum lateinischen Ausdruck ‚sequens‘ (‚[and] the following [page]‘ (und die folgende Seite)), wie er benutzt wird, um den Bereich von zwei Seiten anzugeben, wenn nur die erste Seite genannt wird (zum Beispiel ‚25 sq.‘ oder ‚25 f.‘ anstelle von ‚25–26‘).
- sequentes** Äquivalent zum lateinischen Begriff ‚sequentes‘ (‚[and] the following [pages]‘ (und die folgenden Seiten)), wie er benutzt wird, um einen Bereich mit offenem Ende anzugeben, bei dem nur die erste Seite genannt wird (zum Beispiel ‚25 sqq.‘ oder ‚25 ff.‘).
- passim** Äquivalent zum lateinischen Ausdruck ‚passim‘ (‚throughout‘, ‚here and there‘, ‚scatteredly‘ (durchgehend)).

Andere Ausdrücke, die häufig in Zitaten benutzt werden:

- see** Der Ausdruck ‚see‘ (siehe).
- seealso** Der Ausdruck ‚see also‘ (siehe auch).
- seernote** Ein Ausdruck wie ‚see note ⟨*footnote*⟩‘ oder ‚as in ⟨*footnote*⟩‘ (siehe die Fußnote) wird benutzt, um auf eine frühere Fußnote in einem Zitat hinzuweisen.
- backrefpage** Ein Ausdruck wie ‚see page ⟨*page*⟩‘ (siehe Seite) oder ‚cited on page ⟨*page*⟩‘ (zitiert auf Seite) wird benutzt, um auf frühere Artikel im Literaturverzeichnis hinzuweisen.
- backrefpages** Die Pluralform von backrefpage, zum Beispiel ‚see pages ⟨*pages*⟩‘ oder ‚cited on pages ⟨*pages*⟩‘.
- quotedin** Ein Ausdruck wie ‚quoted in ⟨*citation*⟩‘ (zitiert in ⟨*citation*⟩) wird benutzt, wenn man eine Passage zitieren will, die bereits ein Zitat einer fertigen Arbeit ist.
- citedas** Ein Ausdruck wie ‚henceforth cited as ⟨*shorthand*⟩‘ (zukünftig zitiert als ⟨*Abkürzung*⟩) wird benutzt, um eine Abkürzung eines Zitats einzuleiten.
- thiscite** Der Ausdruck, der in einigen ausführlichen Zitierungsarten verwendet wird, um zwischen den Seitenbereichen der zitierten Artikel (typischerweise ein Artikel in einer Zeitschrift, einer Sammlung oder Tagungsberichten) und den Seitennummern zu unterscheiden, auf die sich die Zitate beziehen. Zum Beispiel: „Author, Title, in: Book, pp. 45–61, thiscite p. 52.“

#### 4.9.2.17. Monatsnamen

- january** Der Name ‚January‘ (Januar).
- february** Der Name ‚February‘ (Februar).
- march** Der Name ‚March‘ (März).
- april** Der Name ‚April‘ (April).

|           |                                   |
|-----------|-----------------------------------|
| may       | Der Name ‚May‘ (Mai).             |
| june      | Der Name ‚June‘ (Juni).           |
| july      | Der Name ‚July‘ (Juli).           |
| august    | Der Name ‚August‘ (August).       |
| september | Der Name ‚September‘ (September). |
| october   | Der Name ‚October‘ (Oktober).     |
| november  | Der Name ‚November‘ (November).   |
| december  | Der Name ‚December‘ (Dezember).   |

#### 4.9.2.18. Namen von Sprachen

|                |                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------|
| langamerican   | Die Sprache ‚American‘ (Amerikanisch) oder ‚American English‘ (amerikanisches Englisch).             |
| langbrazilian  | Die Sprache ‚Brazilian‘ (Brasilianisch) oder ‚Brazilian Portuguese‘ (brasilianisches Portugiesisch). |
| langdanish     | Die Sprache ‚Danish‘ (Dänisch).                                                                      |
| langdutch      | Die Sprache ‚Dutch‘ (Niederländisch).                                                                |
| langenglish    | Die Sprache ‚English‘ (Englisch).                                                                    |
| langfrench     | Die Sprache ‚French‘ (Französisch).                                                                  |
| langgerman     | Die Sprache ‚German‘ (Deutsch).                                                                      |
| langgreek      | Die Sprache ‚Greek‘ (Griechisch).                                                                    |
| langitalian    | Die Sprache ‚Italian‘ (Italienisch).                                                                 |
| langlatin      | Die Sprache ‚Latin‘ (Latein).                                                                        |
| langnorwegian  | Die Sprache ‚Norwegian‘ (Norwegisch).                                                                |
| langportuguese | Die Sprache ‚Portuguese‘ (Portugiesisch).                                                            |
| langspanish    | Die Sprache ‚Spanish‘ (Spanisch).                                                                    |
| langswedish    | Die Sprache ‚Swedish‘ (Schwedisch).                                                                  |

Die folgenden Zeichenketten sind für die Verwendung in Phrasen wie ‚translated from [the] English by ⟨*translator*⟩ (übersetzt aus dem Englischen von ⟨*Übersetzer*)‘ vorgesehen:

|               |                                                                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| fromamerican  | Der Ausdruck ‚from [the] American‘ (aus dem Amerikanischen) oder ‚from [the] American English‘ (aus dem amerikanischen Englischen).             |
| frombrazilian | Der Ausdruck ‚from [the] Brazilian‘ (aus dem Brasilianischen) oder ‚from [the] Brazilian Portuguese‘ (aus dem brasilianischen Portugiesischen). |

|                             |                                                                 |
|-----------------------------|-----------------------------------------------------------------|
| <code>fromdanish</code>     | Der Ausdruck ‚from [the] Danish‘ (aus dem Dänischen).           |
| <code>fromdutch</code>      | Der Ausdruck ‚from [the] Dutch‘ (aus dem Niederländischen).     |
| <code>fromenglish</code>    | Der Ausdruck ‚from [the] English‘ (aus dem Englischen).         |
| <code>fromfrench</code>     | Der Ausdruck ‚from [the] French‘ (aus dem Französischen).       |
| <code>fromgerman</code>     | Der Ausdruck ‚from [the] German‘ (aus dem Deutschen).           |
| <code>fromgreek</code>      | Der Ausdruck ‚from [the] Greek‘ (aus dem Griechischen).         |
| <code>fromitalian</code>    | Der Ausdruck ‚from [the] Italian‘ (aus dem Italienischen).      |
| <code>fromlatin</code>      | Der Ausdruck ‚from [the] Latin‘ (aus dem Lateinischen).         |
| <code>fromnorwegian</code>  | Der Ausdruck ‚from [the] Norwegian‘ (aus dem Norwegischen).     |
| <code>fromportuguese</code> | Der Ausdruck ‚from [the] Portuguese‘ (aus dem Portugiesischen). |
| <code>fromspanish</code>    | Der Ausdruck ‚from [the] Spanish‘ (aus dem Spanischen).         |
| <code>fromswedish</code>    | Der Ausdruck ‚from [the] Swedish‘ (aus dem Schwedischen).       |

**4.9.2.19. Ländernamen** Ländernamen werden lokalisiert, indem man die Zeichenkette `country` mit dem ISO-3166 Ländercode als Schlüssel benutzt. Die kurze Version der Übersetzung sollte der ISO-3166 Ländercode sein. Beachten Sie, dass nur eine kleine Anzahl von Ländernamen standardmäßig definiert ist, hauptsächlich um dieses Schema zu illustrieren. Diese Schlüssel werden in der `location`-Liste von `@patent`-(Patent-)Einträgen benutzt, können aber auch für andere Zwecke verwendet werden.

|                        |                                                                                                                             |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <code>countryde</code> | Der Name ‚Germany‘ (Deutschland), abgekürzt mit DE.                                                                         |
| <code>countryeu</code> | Der Name ‚European Union‘, abgekürzt mit EU.                                                                                |
| <code>countryep</code> | Ähnlich wie <code>countryeu</code> , aber abgekürzt mit EP. Dies ist für <code>patent</code> -(Patent-)Einträge vorgesehen. |
| <code>countryfr</code> | Der Name ‚France‘, abgekürzt mit FR.                                                                                        |
| <code>countryuk</code> | Der Name ‚United Kingdom‘, abgekürzt (gemäß ISO-3166) mit GB.                                                               |
| <code>countryus</code> | Der Name ‚United States of America‘, abgekürzt mit US.                                                                      |

**4.9.2.20. Patente und Patentanträge** Zeichenketten, die im Zusammenhang mit Patenten stehen, werden lokalisiert, indem man den Ausdruck `patent` mit dem ISO-3166 Ländercode benutzt. Beachten Sie, dass nur eine kleine Anzahl von Patentschlüsseln standardmäßig definiert sind, hauptsächlich um dieses Schema zu illustrieren. Diese Schlüssel werden im `type`-Eingabefeld von `@patent`-(Patent-)Einträgen verwendet.

|                     |                                    |
|---------------------|------------------------------------|
| <code>patent</code> | Der Oberbegriff ‚patent‘ (Patent). |
|---------------------|------------------------------------|

- `patentde` Der Ausdruck ‚German patent‘ (deutsches Patent).
- `patenteu` Der Ausdruck ‚European patent‘ (europäisches Patent).
- `patentfr` Der Ausdruck ‚French patent‘ (französisches Patent).
- `patentuk` Der Ausdruck ‚British patent‘ (britisches Patent).
- `patentus` Der Ausdruck ‚U.S. patent‘ (Patent der Vereinigten Staaten).

Patentanträge werden in ähnlicher Weise abgehandelt, indem man die Zeichenkette `patreq` als Basis des Namens des Schlüssels verwendet:

- `patreq` Der Oberbegriff ‚patent request‘ (Patentantrag).
- `patreqde` Der Ausdruck ‚German patent request‘ (deutscher Patentantrag).
- `patreqeu` Der Ausdruck ‚European patent request‘ (europäischer Patentantrag).
- `patreqfr` Der Ausdruck ‚French patent request‘ (französischer Patentantrag).
- `patrequk` Der Ausdruck ‚British patent request‘ (britischer Patentantrag).
- `patrequs` Der Ausdruck ‚U.S. patent request‘ (Patentantrag der Vereinigten Staaten).

## 4.10. Formatierungsbefehle

Dieser Abschnitt bezieht sich auf § 3.8 im Benutzerteil dieses Handbuchs. Die Stile des Literaturverzeichnisses und der Zitate sollten gut mit den Befehlen, die in diesem Abschnitt diskutiert werden, zusammenarbeiten, um einen gewissen Grad an fortgeschrittener Konfigurierbarkeit zu erzielen. Die Benutzer sollten nicht dazu gezwungen sein, neue Stildateien zu schreiben, wenn sie lediglich die Leerzeichensetzung der Literaturverzeichnisse oder die Zeichensetzung in den Zitaten verändern wollen.

### 4.10.1. Frei definierbare Befehle und Haken

Dieser Abschnitt bezieht sich auf § 3.8.1 im Benutzerteil dieses Handbuchs. Die Befehle und Haken, die hier diskutiert werden, sind dafür gedacht, von Benutzern neu definiert zu werden, allerdings haben die Stile des Literaturverzeichnisses und der Zitate meist eine Standardeinstellung, die sich von der Standardeinstellung des Paketes unterscheidet. Diese Befehle werden in `biblatex.def` definiert. Beachten Sie, dass alle Befehle, die mit `\mk. . .` beginnen, ein obligatorisches Argument benötigen.

- `\bibnamedash` Der Bindestrich als Ersatz für sich wiederholende Autoren oder Herausgeber im Literaturverzeichnis. Die Standardeinstellung ist ein ‚em‘- oder ein ‚en‘-Bindestrich, abhängig von der Einrückung der Liste der Bezugnahmen.

- `\labelnamepunct` Der Separator, der nach dem Namen abgedruckt wird, der für die Alphabetisierung des Literaturverzeichnisses benutzt wird (`author` (Autor) oder `editor` (Herausgeber)), falls das `author`-Eingabefeld undefiniert ist. Benutzen Sie hier diesen Separator anstelle von `\newunitpunct`. Die Standardeinstellung ist `\newunitpunct`, das bedeutet, dass es sich in der Handhabung nicht von der regulären Zeichensetzung unterscheidet, aber eine komfortable Rekonfiguration erlaubt.
- `\subtitlepunct` Der Separator, der zwischen den Feldern `title` (Titel) und `subtitle` (Untertitel), `booktitle` (Buchtitel) und `booksubtitle` (Buchuntertitel) als auch `maintitle` (Haupttitel) und `mainsubtitle` (Hauptuntertitel) abgedruckt wird. Benutzen Sie hier diesen Separator anstelle von `\newunitpunct`. Die Standardeinstellung ist `\newunitpunct`, das bedeutet, dass es sich in der Handhabung nicht von der regulären Zeichensetzung unterscheidet, aber eine komfortable Rekonfiguration erlaubt.
- `\intitlepunct` Der Separator, der zwischen dem Wort „in“ bzw. „im“ und dem nachfolgenden Titel in Einträgen wie `@article` (im Artikel), `@inbook` (im Buch), `@incollection` (in der Sammlung) etc. abgedruckt wird. Benutzen Sie hier diesen Separator anstelle von `\newunitpunct`. Die Standardeinstellung ist ein Doppelpunkt mit einem Leerzeichen.
- `\bibpagespunct` Dieser Separator wird vor dem `pages`-(Seiten-)Eingabefeld abgedruckt. Benutzen Sie hier diesen Separator anstelle von `\newunitpunct`. Die Standardeinstellung ist ein Komma mit einem Leerzeichen.
- `\bibpageref` Dieser Separator wird vor dem `pageref`-(Seitenreferenz-)Eingabefeld abgedruckt. Benutzen Sie hier diesen Separator anstelle von `\newunitpunct`. Die Standardeinstellung ist ein Leerzeichen.
- `\multinamedelim` Das Trennzeichen, das zwischen mehreren Einträgen einer Namensliste wie `author` (Autor) oder `editor` (Herausgeber) abgedruckt wird, wenn die Liste mehr als zwei Namen enthält. Wenn nur zwei Namen in der Liste stehen, benutzen Sie `\finalnamedelim`. Dieser Befehl sollte mit allen Formatierungsrichtlinien für Namenslisten funktionieren.
- `\finalnamedelim` Benutzen Sie diesen Befehl anstelle von `\multinamedelim` vor dem letzten Namen in einer Namensliste.
- `\revsdnamedelim` Ein zusätzliches Trennzeichen, das nach dem ersten Namen in einer Namensliste abgedruckt wird (zusätzlich zu `\finalnamedelim`), wenn der erste Name umgekehrt ist. Dieser Befehl sollte in allen Formatierungsrichtlinien für Namenslisten genutzt werden.

- `\andothersdelim` Das Trennzeichen, das vor der lokalisierten Zeichenkette ‚andothers‘ (und andere) abgedruckt wird, wenn eine Namensliste wie `author` (Autor) oder `editor` (Herausgeber) abgekürzt wird. Dieser Befehl sollte mit allen Formatierungsrichtlinien für Namenslisten funktionieren.
- `\multilistdelim` Das Trennzeichen, das zwischen mehrfachen Einträgen in einer Literaturliste wie `publisher` (Verleger) oder `location` (Ort) abgedruckt wird, wenn die Liste mehr als zwei Namen enthält. Wenn nur zwei Namen in der Liste stehen, benutzen Sie `\finalistdelim`. Dieser Befehl sollte mit allen Formatierungsrichtlinien für Literaturlisten funktionieren.
- `\finalistdelim` Nutzen Sie diesen Befehl anstelle von `\multilistdelim` vor dem letzten Eintrag in einer Literaturliste.
- `\andmoredelim` Das Trennzeichen, das vor der lokalisierten Zeichenkette ‚andmore‘ (und weitere) abgedruckt wird, wenn eine Literaturliste wie `publisher` (Verleger) oder `location` (Ort) abgekürzt wird. Dieser Befehl sollte mit allen Formatierungsrichtlinien für Literaturlisten funktionieren.
- `\multicitedelim` Das Trennzeichen, das zwischen Zitaten abgedruckt wird, wenn mehrfache Schlüssel­einträge an einen einzigen Zitierungsbefehl übergeben werden. Dieser Befehl sollte mit allen Definitionen von Zitierungsbefehlen funktionieren, so zum Beispiel im `<sepcode>`-Argument, das an `\DeclareCiteCommand` übergeben wird. Betrachten Sie § 4.3.1 für weitere Details.
- `\supercitedelim` Wie `\multinamedelim`, aber nur für den `\supercite`-Befehl vorgesehen.
- `\compcitedelim` Wie `\multicitedelim`, allerdings für Zitierungsstile vorgesehen, die Mehrfachziti­erungen komprimieren, das heißt, die den Autor nur einmal angeben, wenn anschließende Zitate denselben Autor haben usw.
- `\nametitledelim` Das Trennzeichen, das zwischen dem Autor/Herausgeber und dem Titel abgedruckt wird. Dieser Befehl sollte mit allen Definitionen aller Zitierungsbefehle von Autor und Titel und einigen ausführlichen Zitierungsstilen funktionieren.
- `\nameyeardelim` Das Trennzeichen, das zwischen Autor/Herausgeber und dem Jahr abgedruckt wird. Dieser Befehl sollte mit allen Definitionen aller Zitierungsbefehle von Autor und Jahr funktionieren.
- `\prenotedelim` Das Trennzeichen, das nach dem `<prenote>`-Argument eines Zitierungsbefehls abgedruckt wird.
- `\postnotedelim` Das Trennzeichen, das nach dem `<postnote>`-Argument eines Zitierungsbefehls abgedruckt wird.

`\mkbibnamelast`{*text*}

Haken in der Formatierung für den Nachnamen, für die Verwendung in allen Formatierungsrichtlinien von Namenslisten.

`\mkbibnamefirst`{*text*}

Wie `\mkbibnamelast`, aber für den Vornamen.

`\mkbibnameprefix`{*text*}

Wie `\mkbibnamelast`, aber für das Namenspräfix.

`\mkbibnameaffix`{*text*}

Wie `\mkbibnamelast`, aber für den Namenszusatz.

#### 4.10.2. Sprachspezifische Befehle

Dieser Abschnitt bezieht sich auf § 3.8.2 im Benutzerteil dieses Handbuchs. Die Befehle, die hier diskutiert werden, werden in der Regel von lokalisierten Modulen verarbeitet, können aber auch auf einer per-Sprach-Basis vom Benutzer neu definiert werden. Beachten Sie, dass alle Befehle, die mit `\mk...` beginnen, ein oder mehrere obligatorische Argumente benötigen.

`\bibrangedash` Der sprachspezifische Bereich eines Bindestrichs.

`\bibdatedash` Der sprachspezifische Bereich eines Bindestrichs in einer Datumsangabe.

`\mkbibdatelong` Nimmt die Einträge von drei Eingabefeldern, welche die drei Komponenten eines Datums (in der Reihenfolge Jahr/Monat/Tag) enthalten, und benutzt ihre Werte, um ein Datum im sprachspezifischen langen Format darzustellen.

`\mkbibdateshort` Wie `\mkbibdatelong`, benutzt aber das sprachspezifische kurze Format.

`\finalandcomma` Bildet das Komma ab, das vor dem letzten ‚and‘ (und) in einer Aufzählung eingefügt wird, falls dies in der jeweiligen Sprache verfügbar ist.

`\mkbibordinal`{*integer*}

Nimmt ein ganzzahliges Argument und druckt es als Ordinalzahl.

`\mkbibmascord`{*integer*}

Wie `\mkbibordinal`, druckt aber eine männliche Ordinalzahl, falls dies in der jeweiligen Sprache verfügbar ist.

`\mkbibfemord{⟨integer⟩}`

Wie `\mkbibordinal`, druckt aber eine weibliche Ordinalzahl, falls dies in der jeweiligen Sprache verfügbar ist.

`\mkbibordedition{⟨integer⟩}`

Wie `\mkbibordinal`, aber für die Verwendung mit dem Begriff ‚edition‘ (Ausgabe).

`\mkbibordseries{⟨integer⟩}`

Wie `\mkbibordinal`, aber für die Verwendung mit dem Begriff ‚series‘ (Serie).

#### 4.10.3. Frei definierbare Längen und Zähler

Dieser Abschnitt bezieht sich auf § 3.8.3 im Benutzerteil dieses Handbuchs. Die hier beschriebenen Längenregister und Zähler sind dafür vorgesehen, vom Benutzer verändert zu werden. Die Literaturverzeichnis- und Zitatstile sollten, wo es geht, mit ihnen zusammenarbeiten und können auch eine Standardeinstellung mit sich bringen, die sich vom Standard des Paketes unterscheidet.

`\bibhang` Die überhängende Einrückung des Literaturverzeichnisses. Stile des Literaturverzeichnisses sollten diese Länge in die Definitionen ihrer Literaturverzeichnisumgebung integrieren, falls es möglich ist.

`\biblabelsep` Der horizontale Platz zwischen Einträgen und ihren dazugehörigen Labeln (Beschriftungen). Literaturverzeichnisstile, die `list`-Umgebungen verwenden und ein Label abdrucken, sollten `\labelsep` zu `\biblabelsep` in den Definitionen ihrer jeweiligen Umgebungen ändern.

`\bibitemsep` Der vertikale Platz zwischen den individuellen Einträgen im Literaturverzeichnis. Literaturverzeichnisstile, die `list`-Umgebungen verwenden, sollten `\itemsep` zu `\bibitemsep` in den Definitionen ihrer jeweiligen Umgebungen ändern.

`\bibparsep` Der vertikale Platz zwischen Absätzen innerhalb eines Eintrages in einem Literaturverzeichnis. Literaturverzeichnisstile, die `list`-Umgebungen verwenden und ein Label abdrucken, sollten `\parsep` zu `\bibparsep` in den Definitionen ihrer jeweiligen Umgebungen ändern.

`abbrvpenalty` Der Abstand, der von `\addabbrvspace`, `\addabthinspace` und `\adddotsspace` verwendet wird. Betrachten Sie § 4.7.4 für Details.

`lownamepenalty` Der Abstand, der von `\addlowpenspace` und `\addlpthinspace` verwendet wird. Betrachten Sie § 4.7.4 für Details.

`highnamepenalty` Der Abstand, der von `\addhighpenspace` und `\addhpthinspace` verwendet wird. Betrachten Sie § 4.7.4 für Details.

#### 4.10.4. Zusätzliche Befehle und Haken

Die zusätzlichen Befehle und Funktionen in diesem Abschnitt dienen einem besonderen Zweck. Einige von ihnen werden von `biblatex` verwendet, um auf die eine oder andere Weise mit dem Literaturverzeichnis- und Zitierungsstilen zu kommunizieren.

`\mkbibemph{<text>}`

Ein generischer Befehl, der sein Argument in Kursivschrift abdruckt. Dies ist eine einfache Hülle um den Standard-`\emph`-Befehl. Abgesehen davon benutzt er `\setpunctfont` aus § 4.7.1, um die Schriftart des nächsten Satzzeichens, das dem Text folgt, in Kursivschrift anzupassen. Wenn die `punctfont`-Paketoption nicht aktiviert ist, verhält sich dieser Befehl wie `\emph`.

`\mkbibbold{<text>}`

Ähnliches Konzept wie `\mkbibemph`, druckt den Text allerdings fett. Dies ist eine einfache Hülle um den Standard-`\textbf`-Befehl, welcher `\setpunctfont` beinhaltet. Wenn die `punctfont`-Paketoption nicht aktiviert ist, verhält sich dieser Befehl wie `\textbf`.

`\mkbibquote{<text>}`

Ein generischer Befehl, der sein Argument in Anführungszeichen setzt. Wenn das `csquotes`-Paket geladen ist, benutzt dieser Befehl die sprachspezifischen Anführungszeichen, die vom Paket vorgegeben werden. `\mkbibquote` unterstützt auch die ‚American-style‘-Zeichensetzung. Siehe `\DeclareQuotePunctuation` in § 4.7.5 für weitere Details.

`\mkbibparens{<text>}`

Ein generischer Befehl, der sein Argument in (runde) Klammern setzt. Dieser Befehl ist verschachtelbar. Wenn es verschachtelt ist, wechselt er zwischen runden und eckigen Klammern, je nach Tiefe der Verschachtelung.

`\mkbibbrackets{<text>}`

Ein generischer Befehl, der sein Argument in eckige Klammern setzt. Dieser Befehl ist verschachtelbar. Wenn er verschachtelt ist, wechselt er zwischen eckigen und runden Klammern, je nach Tiefe der Verschachtelung.

`\bibopenparen`*{text}*`\bibcloseparen`

Alternative Syntax für `\mkbibparens`. Dies funktioniert auch in Gruppen. Beachten Sie, dass `\bibopenparen` und `\bibcloseparen` immer ausgeglichen werden müssen.

`\bibopenbracket`*{text}*`\bibclosebracket`

Alternative Syntax für `\mkbibbrackets`. Dies funktioniert auch in Gruppen. Beachten Sie, dass `\bibopenbracket` und `\bibclosebracket` immer ausgeglichen werden müssen.

`\mkbibfootnote`*{text}*}

Ein generischer Befehl, der sein Argument als Fußnote abdruckt. Dies ist eine Hülle um den Standard-LaTeX-`\footnote`-Befehl, welcher überflüssige Leerzeichen vor der Fußnotenmarkierung entfernt und verschachtelte Fußnoten verhindert. In der Standardeinstellung schreibt `\mkbibfootnote` den ersten Buchstaben groß und fügt automatisch einen Punkt am Ende hinzu. Sie können dieses Verhalten ändern, indem Sie `\bibfootnotewrapper`-Makro, welches unten beschrieben ist, neu definieren.

`\mkbibfootnotetext`*{text}*}

Wie `\mkbibfootnote`, benutzt allerdings den `\footnotetext`-Befehl.

`\mkbibendnote`*{text}*}

Ähnliches Konzept wie `\mkbibfootnote`, allerdings druckt es sein Argument als Endnote. `\mkbibendnote` entfernt überflüssige Leerzeichen vor der Endnotenmarkierung und verhindert Verschachtelungen. Er unterstützt sowohl den `\endnote`-Befehl vom `endnotes`-Paket als auch den `\pagenote`-Befehl vom `pagenote`-Paket und die `memoir`-Klasse. Wenn beide Befehle verfügbar sind, entscheidet `\endnote` nach Vorrang. Wenn keine Unterstützung von ‚endnote‘ verfügbar ist, gibt `\mkbibendnote` einen Fehler an und fällt zurück auf `\footnote`. In der Standardeinstellung schreibt `\mkbibendnote` den ersten Buchstaben groß und fügt automatisch einen Punkt am Ende hinzu. Sie können dieses Verhalten ändern, indem Sie `\bibendnotewrapper`-Makro, welches unten beschrieben ist, neu definieren.

`\mkbibendnotetext`*{text}*}

Wie `\mkbibendnote`, benutzt allerdings den `\endnotetext`-Befehl. Bitte beachten Sie, dass zum Zeitpunkt, als dies geschrieben wurde, weder das `daspagenote`-Paket noch die `memoir`-Klasse einen entsprechenden `\pagenotetext`-Befehl liefert. In diesem Fall zeigt `\mkbibendnote` einen Fehler an und fällt zurück zu `\footnotetext`.

`\bibfootnotewrapper{<text>}`

Eine innere Hülle, die das `<text>`-Argument von `\mkbibfootnote` und `\mkbibfootnotetext` umschließt. Beispielsweise reduziert sich `\mkbibfootnote` letztendlich auf:

```
\footnote{\bibfootnotewrapper{text}}
```

Die Hülle sorgt für die Großschreibung am Anfang und fügt einen Punkt am Ende hinzu. Die Standardeinstellung ist:

```
\newcommand{\bibfootnotewrapper}[1]{\bibsentence #1\addperiod}
```

Wenn Sie keine Großschreibung am Anfang oder einen Punkt am Ende wünschen, modifizieren Sie nicht `\mkbibfootnote`, sondern definieren Sie stattdessen `\bibfootnotewrapper` neu.

`\bibendnotewrapper{<text>}`

Ähneln vom Konzept `\bibfootnotewrapper`, ist allerdings verwandt mit den Befehlen `\mkbibendnote` und `\mkbibendnotetext`.

`\mkbibsuperscript{<text>}`

Ein generischer Befehl, welcher den Parameter als hochgestellten Text ausgibt. Dies ist eine einfache Hülle, um das Standardkommando `\textsuperscript` von LaTeX, welche falsche Leerräume entfernt und die Silbentrennung des vorangegangenen Wortes erlaubt.

`\mkbibmonth{<integer>}`

Dieses Kommando nimmt einen Integer-Wert als Parameter und gibt diesen als Monatsnamen aus. Obwohl die Ausgabe des Befehls abhängig von der Sprache ist, so ist die Definition des Kommandos davon nicht betroffen. Somit ist es im Normalfall nicht umdefiniert in Lokalisationsmodulen.

`\mkdatezeros{<integer>}`

Dieser Befehl entfernt führende Nullen von einer Zahl oder schützt diese, abhängig von der `datezeros`-Paketoption (§ 3.1.2.1). Der Befehl kann in der Definition von Makros genutzt werden, die ein Datum formatieren.

`\stripzeros{<integer>}`

Dieses Kommando entfernt führende Nullen von einer Zahl. Es kann für das Formatieren oder Aufzählen von Daten genutzt werden.

- `shorthandwidth` Eine besondere, Feld formatierende Anweisung, die intern von biblatex genutzt wird. Wenn die bibliographischen Daten von der bbl-Datei gelesen werden, misst biblatex die Werte von allen shorthand-Feldern und setzt das Längenregister `\shorthandwidth` auf den Wert des weitesten Stenogramms (siehe § 4.10.5). Um die korrekte Weite zu bestimmen, nutzt das Paket zwei Faktoren: die Definition von `\bibfont` und seine Formatierungsanweisung. Alle Stilarten sollten diese Anweisung so einstellen, dass sie mit dem Format der Stichwortlisten zusammenpassen.
- `labelnumberwidth` Dieser Befehl ähnelt `shorthandwidth`, bezieht sich aber auf das Feld `labelnumber` und das Längenregister `\labelnumberwidth`. Numerische Stile sollten diesen Befehl so einrichten, dass er zum Format der Bibliographie passt.
- `labelalphawidth` Dieser Befehl ähnelt `shorthandwidth`, bezieht sich auf auf das Feld `labelalpha` und das Längenregister `\labelalphawidth`. Alphabetische Stile sollten diesen Befehl so einrichten, dass er zum Format der Bibliographie passt.
- `bibhyperref` Ein besonderer Befehl zur Formatierung, welcher zusammen mit `\printfield` und `\printtext` genutzt wird. Er umhüllt seinen Parameter mit dem Kommando `\bibhyperref`. Siehe § 4.6.4 für Details.
- `bibhyperlink` Ein besonderer Befehl zur Formatierung, welcher zusammen mit `\printfield` und `\printtext` genutzt wird. Er umhüllt seinen Parameter mit dem Kommando `\bibhyperlink`. Siehe § 4.6.4 für Details. Der Parameter `<name>`, welcher an `\bibhyperlink` weitergegeben wird, ist der Wert des Feldes `entrykey`.
- `bibhypertarget` Ein besonderer Befehl zur Formatierung, welcher zusammen mit `\printfield` und `\printtext` genutzt wird. Er umhüllt seinen Parameter mit dem Kommando `\bibhypertarget`. Siehe § 4.6.4 für Details. Der Parameter `<name>`, welcher an `\bibhypertarget` weitergegeben wird, ist der Wert des Feldes `entrykey`.

#### 4.10.5. Hilfreiche Längen, Zähler, und andere Features

Die Längenregister und Zähler (counter), welche hier besprochen werden, nutzt biblatex, um Informationen an die Bibliographie und die Zitierungsstile weiterzugeben. Stellen Sie sich diese als Register vor, die nur gelesen werden können. Beachten Sie außerdem, dass alle Zähler LaTeX-Counter sind. Nutzen Sie `\value{counter}` um den aktuellen Wert zu erfahren.

- `\shorthandwidth` Dieses Längenregister zeigt die Weite des breitesten Objektes in `shorthand` an. Die Bibliographie-Stile sollten diese Länge in die Definition der Stichwortlisten einbinden, sofern anwendbar.
- `\labelnumberwidth` Dieses Längenregister zeigt die Weite des breitesten Objektes in `labelnumber` an. Numerische Bibliographie-Stile sollten diese Länge in die Definition der Bibliographie-Umgebung einbinden.

|                               |                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\labelalphawidth</code> | Dieses Längenregister zeigt die Weite des breitesten Objektes in <code>labelalpha</code> an. Alphabetische Bibliographie-Stile sollten diese Länge in die Definition der Bibliographie-Umgebung einbinden.                                                                                                                                                                                 |
| <code>maxextraalpha</code>    | Dieser Zähler entspricht der höchsten Nummer, die in einem <code>extraalpha</code> -Feld gefunden wurde.                                                                                                                                                                                                                                                                                   |
| <code>maxextrayear</code>     | Dieser Zähler entspricht der höchsten Nummer, die in einem <code>extrayear</code> -Feld gefunden wurde.                                                                                                                                                                                                                                                                                    |
| <code>refsection</code>       | Dieser Zähler zeigt die momentane <code>refsection</code> -Umgebung an. Wird in einer Überschrift der Bibliographie abgefragt, so gibt der Zähler den Wert der Option <code>refsection</code> zurück, welcher an <code>\printbibliography</code> weitergegeben wird.                                                                                                                       |
| <code>refsegment</code>       | Dieser Zähler zeigt die momentane <code>refsegment</code> -Umgebung an. Wird in einer Überschrift der Bibliographie abgefragt, so gibt der Zähler den Wert der Option <code>refsegment</code> zurück, welcher an <code>\printbibliography</code> weitergegeben wird.                                                                                                                       |
| <code>maxnames</code>         | Dieser Zähler gibt die Einstellung der Paketooption <code>maxnames</code> zurück.                                                                                                                                                                                                                                                                                                          |
| <code>minnames</code>         | Dieser Zähler gibt die Einstellung der Paketooption <code>minnames</code> zurück.                                                                                                                                                                                                                                                                                                          |
| <code>maxitems</code>         | Dieser Zähler gibt die Einstellung der Paketooption <code>maxitems</code> zurück.                                                                                                                                                                                                                                                                                                          |
| <code>minitems</code>         | Dieser Zähler gibt die Einstellung der Paketooption <code>minitems</code> zurück.                                                                                                                                                                                                                                                                                                          |
| <code>instcount</code>        | Dieser Zähler wird von <code>biblatex</code> sowohl für jedes Zitat, als auch für jeden Eintrag in der Bibliographie und im Stenogramm erhöht. Der Wert dieses Zählers steht für genau eine Instanz einer Referenz im Dokument.                                                                                                                                                            |
| <code>citetotal</code>        | Dieser Zähler ist nur im <code>\loopcode</code> eines Zitatbefehls vorhanden, welcher durch <code>\DeclareCiteCommand</code> definiert wurde. Er gibt die Gesamtzahl der gültigen Eintragungsschlüssel zurück, die an den Zitatbefehl gereicht werden.                                                                                                                                     |
| <code>citecount</code>        | Dieser Zähler ist nur im <code>\loopcode</code> eines Zitatbefehls vorhanden, welcher durch <code>\DeclareCiteCommand</code> definiert wurde. Er gibt die Nummer des Eintragungsschlüssels wieder, welcher im Moment durch den <code>\loopcode</code> verarbeitet wird.                                                                                                                    |
| <code>multicitetotal</code>   | Dieser Zähler ähnelt <code>citetotal</code> , ist allerdings nur in <code>multicite</code> -Befehlen verfügbar. Er enthält die Gesamtzahl der Zitate, die an das <code>multicite</code> -Kommando gereicht werden. Beachten Sie, dass jede dieser Zitierungen mehr als einen Eintragungsschlüssel enthalten kann. Diesbezügliche Informationen liefert der Zähler <code>citetotal</code> . |
| <code>multicitecount</code>   | Dieser Zähler ähnelt <code>citecount</code> , ist allerdings nur in <code>multicite</code> -Befehlen verfügbar. Er enthält die Nummer der Zitierung, die gerade verarbeitet wird. Beachten Sie, dass jede dieser Zitierungen mehr als einen Eintragungsschlüssel enthalten kann. Diesbezügliche Informationen liefern die Zähler <code>citetotal</code> und <code>citecount</code> .       |

- `listtotal` Dieser Zähler enthält die Gesamtzahl von Elementen in der momentan verwendeten Liste. Sein eigentlicher Nutzen ist die Verwendung innerhalb von Formatierungsbefehlen für Listen. Als Ausnahme für diese Regel sei die Verwendung im zweiten optionalen Parameter von `\printnames` und `\printlist` zu nennen. Siehe § 4.4.1 für weitere Details. Für jede Liste existiert ein gleichnamiger Zähler, welcher die Gesamtzahl von Elementen in seiner Liste enthält. Beispielsweise drückt `author` die Zahl von Elementen der `author`-Liste aus. Dies gilt sowohl für Namenslisten, als auch für Listen von Literalen. Diese Zähler ähneln `listtotal`, mit dem Unterschied, dass sie auch unabhängig von Formatierungsbefehlen für Listen angewandt werden können. Beispielsweise könnte ein Bibliographie-Stil den Zähler `editor` überprüfen, um zu entscheiden, ob der Begriff "editor", oder besser sein Plural "editors" hinter der Liste der Herausgeber erscheinen soll.
- `listcount` Dieser Zähler trägt die Nummer des Listenobjekts, welches gerade verarbeitet wird. Er kann für Befehle zur Listenformatierung genutzt werden. Für den Einsatz an anderer Stelle trägt der Zähler keinen sinnvollen Wert.
- `liststart` Dieser Zähler trägt den Parameter `<start>`, der an `\printnames` oder `\printlist` weitergegeben wird. Er kann für Befehle zur Listenformatierung genutzt werden. Für den Einsatz an anderer Stelle trägt der Zähler keinen sinnvollen Wert.
- `liststop` Dieser Zähler trägt den Parameter `<stop>`, der an `\printnames` oder `\printlist` weitergegeben wird. Er kann für Befehle zur Listenformatierung genutzt werden. Für den Einsatz an anderer Stelle trägt der Zähler keinen sinnvollen Wert.
- `\currentfield` Dieser Befehl gibt den Namen des Feldes zurück, welches gerade von `\printfield` verarbeitet wird. Diese Information ist nur lokal in Befehlen zur Feldformatierung verfügbar.
- `\currentlist` Dieser Befehl gibt den Namen des Literals zurück, welches gerade von `\printlist` verarbeitet wird. Diese Information ist nur lokal in Befehlen zur Listenformatierung verfügbar.
- `\currentname` Dieser Befehl gibt den Namen der Namensliste zurück, welche gerade von `\printnames` verarbeitet wird. Diese Information ist nur lokal in Befehlen zur Namesformatierung verfügbar.

#### 4.10.6. Mehrzweckaufhänger

`\AtBeginBibliography{<code>}`

Hängt den `<code>` an einen internen Aufhänger, der am Anfang der Bibliographie ausgeführt wird. Der `<code>` wird zu Beginn der Referenzlisten ausgeführt, direkt nach dem `<begin code>` von `\defbibenvironment`. Dieser Befehl kann nur in der Präambel genutzt werden.

`\AtBeginShorthands{<code>}`

Hängt den `<code>` an einen internen Aufhänger, der am Anfang der Stenogramme ausgeführt wird. Der `<code>` wird zu Beginn der Stenogrammlisten ausgeführt, direkt nach dem `<begin code>` von `\defbibenvironment`. Dieser Befehl kann nur in der Präambel genutzt werden.

`\AtEveryBibitem{<code>}`

Hängt den `<code>` an einen internen Aufhänger, der zu Beginn jedes Elementes der Bibliographie ausgeführt wird. Der `<code>` wird direkt nach dem `<item code>` von `\defbibenvironment` ausgeführt. Die bibliographischen Daten des entsprechenden Eintrages sind an dieser Stelle verfügbar. Dieser Befehl kann nur in der Präambel genutzt werden.

`\AtEveryLositem{<code>}`

Hängt den `<code>` an einen internen Aufhänger, der zu Beginn jedes Elementes der Stenogrammlisten ausgeführt wird. Der `<code>` wird direkt nach dem `<item code>` von `\defbibenvironment` ausgeführt. Die bibliographischen Daten des entsprechenden Eintrages sind an dieser Stelle verfügbar. Dieser Befehl kann nur in der Präambel genutzt werden.

`\AtEveryCite{<code>}`

Hängt den `<code>` an einen internen Aufhänger, der zu Beginn jedes Zitierungskommandos ausgeführt wird. Der `<code>` wird direkt vor dem `<precode>` des Kommandos ausgeführt (siehe § 4.3.1). Es sind an dieser Stelle keinerlei bibliographischen Daten verfügbar. Dieser Befehl kann nur in der Präambel genutzt werden.

`\AtEveryCitekey{<code>}`

Hängt den `<code>` an einen internen Aufhänger, der für jeden Eintragsschlüssel, der man an ein Zitierungskommando weiterreicht, einmal ausgeführt wird. Der `<code>` wird direkt vor dem `<loopcode>` des Kommandos ausgeführt (siehe § 4.3.1). Die bibliographischen Daten des entsprechenden Eintrages sind an dieser Stelle verfügbar. Dieser Befehl kann nur in der Präambel genutzt werden.

`\AtNextCite{<code>}`

Ähnelt `\AtEveryCite`, betrifft aber nur den nächsten Zitierungsbefehl. Der interne Aufhänger wird nach einmaliger Ausführung gelöscht. Dieser Befehl kann innerhalb des Dokumentes genutzt werden.

`\AtNextCitekey{<code>}`

Ähneln `\AtEveryCite`, betrifft aber nur den nächsten Eintragsschlüssel. Der interne Aufhänger wird nach einmaliger Ausführung gelöscht. Dieser Befehl kann innerhalb des Dokumentes genutzt werden.

`\AtDataInput[<entrytype>]{<code>}`

Hängt den `<code>` an einen internen Aufhänger, der einmal für jeden Eintrag der bibliographischen Daten ausgeführt wird, die aus der `bb1`-Datei importiert werden. Der `<entrytype>` ist der Eintragstyp, auf den der `<code>` angewandt wird. Soll er auf alle Eintragstypen angewandt werden, so kann der optionale Parameter weggelassen werden. Der `<code>` wird direkt nach dem Import des Eintrags ausgeführt. Dieser Befehl kann nur in der Präambel genutzt werden. Sollte es `refsection`-Umgebungen im Dokument geben und ein Eintrag in mehr einer davon zitiert werden, so wird der `<code>` mehrfach für diesen Eintrag ausgeführt. Die Anzahl der entsprechenden Referenzbereiche wird während des Transportes vom Zähler `refsection` beschrieben.

## 4.11. Hinweise und Warnungen

Dieser Abschnitt liefert einige zusätzliche Hinweise bezüglich des Autoren-Interfaces dieses Pakets. Er adressiert außerdem häufig auftauchende Probleme und eventuelle Missverständnisse.

### 4.11.1. Eintragsmengen

Eintragsmengen wurden bereits in § 3.10.5 vorgestellt. Dieser Abschnitt beschäftigt sich mit dem Verarbeiten von Eintragsmengen in einem Bibliographie-Stil. Aus Sicht des Treibers gibt es keinen Unterschied zwischen statischen und dynamischen Eintragsmengen. Beide Typen werden auf gleiche Weise gehandhabt. Sie werden im Normalfall den Befehl `\entryset` aus § 4.4.1 nutzen, um über alle Elemente der Menge zu gehen (in der Reihenfolge, in der sie im Feld `entryset` des `@set`-Eintrags aufgeführt sind, oder aber in der Reihenfolge, in der sie an `\defbibentryset` weitergegeben werden) und `\finentry` an das Ende anzuhängen. Mehr ist nicht zu tun. Das Formatieren wird von den Treibern der jeweiligen Eintragstypen der einzelnen Mengenelemente übernommen:

```
\DeclareBibliographyDriver{set}{%
\entryset{}{}}% \finentry}
```

Ihnen ist vielleicht aufgefallen, dass die numerischen-Stile, welche mit diesem Paket mitgeliefert werden, die Unterteilung von Eintragsmengen unterstützen. Das bedeutet, dass die Mitglieder einer Menge mit einem Buchstaben oder auf andere Weise markiert sind. Zitierungen können sich somit auf die gesamte Menge, oder aber auf

das spezielle Element beziehen. Die Markierungen werden vom bibliographischen Stil wie folgt generiert:

```
\DeclareBibliographyDriver{set}{%
\entryset
{\printfield{entrysetcount}%
\setunit*{\addnbspace}} {}%
\finentry}
```

Das Feld `entrysetcount` trägt einen Ganzzahlwert, der die Position eines Elements der Eintragsmenge repräsentiert. Die Umwandlung dieser Zahl zu einem Buchstaben oder einer anderen Markierung erfolgt über den Formatierungsbefehl von `entrysetcount`. Der Treiber muss dieses Feld ausgeben und einige Leerzeichen einfügen oder aber eine neue Zeile beginnen. Die Ausgabe der Markierungen in Zitierungen erfolgt auf einem ähnlichen Weg. An Stellen, wo in numerischen Stilen `\printfield{labelnumber}` aufgerufen wird, wird einfach das Feld `entrysetcount` angehängen:

```
\printfield{labelnumber}\printfield{entrysetcount}
```

Da dieses Feld nur dann definiert ist, wenn verarbeitete Zitierungen sich auf ein Element der Mengen beziehen, sind zusätzliche Tests nicht nötig.

#### 4.11.2. Informationen zu elektronischen Veröffentlichungen

Die üblichen Stile besitzen Support für Archiv-Referenzen. Der Support für anderen Ressourcen kann einfach hinzugefügt werden. Die Standardstile handhaben das Feld `eprint` wie folgt:

```
\iffieldundef{eprinttype}
{\printfield{eprint}} {\printfield[eprint:\strfield{eprinttype}]{eprint}}
```

Wenn ein `eprinttype`-Feld verfügbar ist, versucht der obige Code, das Feldformat `eprint:⟨eprinttype⟩` zu nutzen. Wenn dieses Format nicht definiert ist, fällt `\printfield` automatisch auf das Feldformat `eprint` zurück. Es gibt zwei vordefinierte Feldformate: das typenspezifischen Format `eprint:arxiv` und das Ersatzformat `eprint`:

```
\DeclareFieldFormat{eprint}{...}
\DeclareFieldFormat{eprint:arxiv}{...}
```

Mit anderen Worten: Den Support für zusätzliche Resources hinzuzufügen ist genauso leicht, wie ein Feldformat namens `eprint:⟨resource⟩` zu definieren, in dem `⟨resource⟩` eine Identifizierung ist, die im Feld `eprinttype` genutzt wird.

### 4.11.3. Externe Abstracts und Annotationen

Externe Abstracts und Annotationen wurden bereits in § 3.10.7 besprochen. Dieser Teil liefert etwas mehr Hintergrundwissen für Stil-Autoren. Die Standardstile nutzen die folgenden Makros (von `biblatex.def`), um Abstracts und Annotationen zu handhaben:

```
\newbibmacro*{annotation}{%
\iffieldundef{annotation}
{\printfile[annotation]{\bibannotationprefix\thefield{entrykey}.tex}}%
{\printfield{annotation}}}
\newcommand*{\bibannotationprefix}{bibannotation-}

\newbibmacro*{abstract}{%
\iffieldundef{abstract}
{\printfile[abstract]{\bibabstractprefix\thefield{entrykey}.tex}}%
{\printfield{abstract}}}\newcommand*{\bibabstractprefix}{bibabstract-}
```

Wenn `abstract` oder `annotation` undefiniert sind, versucht das obige Beispiel die Abstracts/Annotationen von einer externen Datei zu laden. Der Befehl `\printfile` nimmt auch Dateinamen-Präfixe auf, welche eventuell vom Benutzer undefiniert wurden. Beachten Sie, dass Sie `\printfile` explizit aktivieren müssen, indem Sie die Paketoptionen von `loadfiles` setzen (§ 3.1.2.1). Dieses Feature ist aus Performance-Gründen im Normalfall deaktiviert.

### 4.11.4. Disambiguierung der Namen

Biber only

Die `uniquename-` und `unique-list-`Optionen, eingeführt in § 3.1.2.3, unterstützen verschiedene Betriebsmodi. Dieser Abschnitt erklärt die Unterschiede von diesen Optionen mit Beispielen. Die `uniquename-`Option disambiguiert individuelle Namen in der `labelname-`liste. Die `unique-list-`Option disambiguiert die `labelname-`Liste, wenn die Mehrdeutigkeit nach der `maxnames/minnames-`Truncation entstanden ist. Sie können entweder die Option alleine verwenden oder beides kombinieren.

**4.11.4.1. Einzelnamen (`uniquename`)** Wir beginnen mit einigen Beispielen von `uniquename` (Einzelnamen). Betrachten Sie die folgenden Daten:

```
John Doe 2008
Edward Doe 2008
John Smith 2008
Jane Smith 2008
```

Nehmen wir an, wir benutzen einen `author-year-`Stil und setzen `uniquename=false`. In diesem Fall würden wir die folgenden Zitierungen bekommen.

Doe 2008a  
 Doe 2008b  
 Smith 2008a  
 Smith 2008b

Wenn die Familiennamen ambig sind und alle Arbeiten im selben Jahr veröffentlicht wurden, wird ein extra Kürzel an das Jahr angehängt, um die Zitate eindeutig zu machen. Jedoch viele Stilbeschreibungen erzwingen die extra Buchstaben nur für die Disambiguierung für Arbeiten desselben Autors, nicht für Arbeiten von Autoren mit demselben Familiennamen. Um Nachnamen zu disambiguieren wird erwartet, dass Sie weitere Namensteile als Initialen oder in Vollform hinzufügen. Diese Anforderung wird adressiert mit der `uniquename`-Option. Hier sind die gleichen Zitate mit `uniquename=init`:

J. Doe 2008  
 E. Doe 2008  
 Smith 2008a  
 Smith 2008b

`uniquename=init` beschränkt die Namensdisambiguierung auf Initiale. Da ‚J. Smith‘ immer noch ambig ist, wurde kein Namensteil für die ‚Smiths‘ hinzugefügt. Mit `uniquename=full` werden die Namen in voller Länge gedruckt, wenn erforderlich:

J. Doe 2008  
 E. Doe 2008  
 John Smith 2008  
 Jane Smith 2008

Um den Unterschied zwischen `uniquename=init/full` und `allinit/allfull` zu illustrieren, müssen wir einen ‚visible‘-Namen einfügen. Im Folgenden sind ‚visible‘-Namen alle Namen in einer Position vor dem `maxnames/minnames/unique-list-truncation`-punkt. Beispielsweise ergibt dies:

William Jones/Edward Doe/Jane Smith  
 John Doe  
 John Smith

Und `maxnames=1, minnames=1, uniquename=init/full` würde folgende Namenszitate ergeben:

Jones et al.  
 Doe  
 Smith

Disambiguierte Namen, `uniquename=init/full`, berücksichtigen nur die sichtbaren Namen. Alle sichtbaren Familiennamen in diesem Beispiel sind distinkt, keine weiteren Namensteile wurden hinzugefügt. Vergleichen wir die Ausgabe von `uniquename=allinit`:

```
Jones et al.
J. Doe
Smith
```

`allinit` behält alle Namen in allen `labelname`-listen, einschließlich derjenigen, die versteckt und mit ‚et al.‘ ersetzt wurden, die Liste ist abgeschnitten. In dem Beispiel ist ‚John Doe‘ disambiguiert von ‚Edward Doe‘. Da die Mehrdeutigkeit der zwei ‚Smiths‘ nicht durch das Hinzufügen von Initialen aufgelöst werden kann, wurden in diesem Fall keine Initialen hinzugefügt. Nun wollen wir den Output von `uniquename=allfull` mit der Disambiguierung von ‚John Smith‘ mit ‚Jane Smith‘ vergleichen:

```
Jones et al.
J. Doe
John Smith
```

Die Optionen `uniquename=mininit/minfull` sind ähnlich mit `init/full`, sie betrachten nur sichtbare Namen, aber sie erzielen nur minimale Disambiguierung. Dies heißt, sie werden Einzelnamen nur disambiguieren, wenn sie in identischen Listen von Familiennamen auftreten. Betrachten Sie die folgenden Daten:

```
John Doe/William Jones
Edward Doe/William Jones
John Smith/William Edwards
Edward Smith/Allan Johnson
```

Mit `uniquename=init/full` würden wir bekommen:

```
J. Doe and Jones
E. Doe and Jones
J. Smith and Edwards
E. Smith and Johnson
```

Mit `uniquename=mininit/minfull`:

```
J. Doe and Jones
E. Doe and Jones
Smith and Edwards
Smith and Johnson
```

Die ‚Smiths‘ wurden nicht disambiguiert, weil die sichtbaren Namenslisten nicht mehrdeutig sind und die `mininit/minfull`-Optionen dazu dienen, Namen zu disambiguieren, die in nur in identischen Listen vorkommen. Ein anderer Weg ist, dass sie global disambiguieren die Nachnamenliste. Wenn es zu mehrdeutigen Listen kommt, beachten Sie, dass eine verkürzte Liste anzusehen ist, als verschieden von einer anderen verkürzten, selbst wenn die sichtbaren Namen identisch sind. Betrachten Sie als Beispiel die folgenden Daten:

```
John Doe/William Jones
Edward Doe
```

Mit `maxnames=1, uniquename=init/full` würden wir bekommen:

```
J. Doe et al.
E. Doe
```

Mit `uniquename=mininit/minfull`:

```
Doe et al.
Doe
```

Weil sich die Listen in ‚et al.‘ unterscheiden, sind die Namen nicht disambiguiert.

**4.11.4.2. Namenslisten (`uniquelist`)** Mehrdeutigkeit ist auch ein Problem für die Namenslisten. Wenn der `labelname`-Liste die `maxnames/minnames`-Optionen abgeschnitten werden, kann es zu Mehrdeutigkeit kommen. Diese Art von Ambiguität wird von der `uniquelist`-Option adressiert. Beachten Sie die folgenden Daten:

```
Doe/Jones/Smith 2005
Smith/Johnson/Doe 2005
Smith/Doe/Edwards 2005
Smith/Doe/Jones 2005
```

Viele `author-year`-Stile verkürzen lange `author/editor`-Listen in Zitaten. Mit `maxnames=1` würden wir beispielsweise erhalten:

```
Doe et al. 2005
Smith et al. 2005a
Smith et al. 2005b
Smith et al. 2005c
```

Da die Autorennamen nach dem Abschneiden mehrdeutig sind, ist ein extra Kürzel an das Jahr angefügt, um eindeutige Zitate zu gewährleisten. Auch hier haben wir wieder, viele Stilbeschreibungen machen es verbindlich, dass zusätzliche Buchstaben verwendet werden, um Werke von dem gleichen Autor eindeutig zu machen. Um

die Autorenliste zu disambiguieren, ist es in der Regel erforderlich, weitere Namen hinzuzufügen, hinausgehend über den `maxnames/minnames`-Truncationspunkt. Das `uniquelist`-Merkmal adressiert dieses Merkmal. Mit `uniquelist=true` würden wir bekommen:

```
Doe et al. 2005
Smith, Johnson et al. 2005
Smith, Doe and Edwards 2005
Smith, Doe and Jones 2005
```

Die `uniquelist`-Option überschreibt `maxnames/minnames` auf einer pro Eintrag Basis. Im Grunde, ist das, was passiert, dass der ‚et al.‘ Zitateil expandierte bis zum Punkt der Eindeutigkeit – aber nicht weiter darüber hinaus. `uniquelist` kann also mit `uniquename` kombiniert werden. Betrachten Sie die folgenden Daten:

```
John Doe/Allan Johnson/William Jones 2009
John Doe/Edward Johnson/William Jones 2009
John Doe/Jane Smith/William Jones 2009
John Doe/John Smith/William Jones 2009
John Doe/John Edwards/William Jones 2009
John Doe/John Edwards/Jack Johnson 2009
```

Mit `maxnames=1`:

```
Doe et al. 2009a
Doe et al. 2009b
Doe et al. 2009c
Doe et al. 2009d
Doe et al. 2009e
Doe et al. 2009f
```

Mit `maxnames=1, uniquename=full, uniquelist=true`:

```
Doe, A. Johnson et al. 2009
Doe, E. Johnson et al. 2009
Doe, Jane Smith et al. 2009
Doe, John Smith et al. 2009
Doe, Edwards and Jones 2009
Doe, Edwards and Johnson 2009
```

Mit `uniquelist=minyear` geschieht die Disambiguierung nur, wenn die sichtbare Liste identisch ist mit der anderen sichtbaren Liste mit dem gleichen `labelyear`. Dies ist nützlich für `author-year`-Stile, die nur verlangen, dass das Zitat eindeutig ist, jedoch nicht eindeutige Informationen zu Urheberschaften in Zitaten garantieren. Dieser Modus ist konzeptuell mit `uniquename=mininit/minfull` verwandt. Betrachten Sie dieses Beispiele:

Smith/Jones 2000  
Smith/Johnson 2001

Mit `maxnames=1` and `uniquelist=true` würden wir erhalten:

Smith and Jones 2000  
Smith and Johnson 2001

Mit `uniquelist=minyear`:

Smith et al. 2000  
Smith et al. 2001

Mit `uniquelist=minyear` ist nicht klargestellt, dass die Autoren für die beiden Werke verschieden sind, aber die Zitate als Ganzes sind immer noch eindeutig, aber nach dem Jahr ist es anders. Im Gegensatz dazu, `uniquelist=true` disambiguiert die Urheberschaft, auch wenn diese Information nicht erforderlich ist, um die Werke eindeutig in der Bibliographie zu lokalisieren. Betrachten wir ein anderes Beispiel:

Vogel/Beast/Garble/Rook 2000  
Vogel/Beast/Tremble/Bite 2000  
Vogel/Beast/Acid/Squeeze 2001

Mit `maxnames=3`, `minnames=1`, `uniquelist=true` würden wir erhalten:

Vogel, Beast, Garble et al. 2000  
Vogel, Beast, Tremble et al. 2000  
Vogel, Beast, Acid et al. 2001

Mit `uniquelist=minyear`:

Vogel, Beast, Garble et al. 2000  
Vogel, Beast, Tremble et al. 2000  
Vogel et al. 2001

In dem letzten Zitat überschreibt `uniquelist=minyear` nicht `maxnames/minnames`, so muss das Zitat nicht von den beiden anderen disambiguiert werden, weil das Jahr verschieden ist.

#### 4.11.5. Tracker in Gleitkommazahlen und TOC/LOT/LOF

Ist eine Zitierung als Gleitkommazahl gegeben (typischer Weise im Titel einer Abbildung oder einer Tabelle), so werden akademische Rückreferenzen wie ‚ibidem‘ oder Rückreferenzen, die auf dem Seitentracker basieren uneindeutig. Der Grund dafür ist, dass Gleitkommazahlen Objekte sind, die außerhalb des Textflusses platziert werden. Daher lässt sich die Logik solcher Referenzen nur schwer auf sie anwenden. Um Mehrdeutigkeiten zu vermeiden, werden die Zitierungs- und Seitentracker temporär in allen Gleitkommazahlen abgeschaltet. Zusätzlich dazu werden diese Tracker und die Rückreferenztracker (`backref`) im Inhaltsverzeichnis (TOC), Abbildungsverzeichnis (LOF) und dem Tabellenverzeichnis (LOT) abgestellt.

#### 4.11.6. Vermischen von Programmierinterfaces

Das Paket `biblatex` bietet zwei wesentliche Programmierinterfaces um Autorenangaben zu gestalten. Der Befehl `\DeclareBibliographyDriver`, welcher die Handhabbarkeit eines Eintragstyps definiert, wird üblicherweise in `bbx`-Dateien genutzt. Das Kommando `\DeclareCiteCommand`, welches ein neues Zitierungskommando definiert, wird in der Regel in `cbx`-Dateien genutzt. Jedoch ist es in einigen Fällen ratsam, diese beiden Interfaces zu mischen. Beispielsweise gibt der Befehl `\fullcite` eine langatmige Zitierung aus, die dem kompletten Bibliographieeintrag nahe kommt. Der Befehl ist im Grunde wie folgt definiert:

```
\DeclareCiteCommand{\fullcite}
{...}
{\usedriver{...}{\thefield{entrytype}}}
{...}
{...}
```

Wie zu sehen ist, führt der Hauptkode (der die Zitierungen ausgibt) einfach den Treiber der Bibliographie aus, welcher durch `\DeclareBibliographyDriver` für den Typ des momentanen Eintrags definiert ist. Wenn man einen Zitierungsstil für ein langatmiges Zitierungsschemata schreibt, ist es oft sinnvoll die folgende Struktur zu nutzen:

```
\ProvidesFile{example.cbx}[2007/06/09 v1.0 biblatex citation style]

\DeclareCiteCommand{\cite} {...}
{\usedriver{...}{cite:\thefield{entrytype}}} {...} {...}

\DeclareBibliographyDriver{cite:article}{...}
\DeclareBibliographyDriver{cite:book}{...}
\DeclareBibliographyDriver{cite:inbook}{...}
...
```

Ein anderes Beispiel, bei dem es nützlich ist, die Interfaces zu vermischen, stellen Stile mit Überkreuz-Referenzen innerhalb der Bibliographie dar. Beispielsweise werden beim Ausgeben des Eintrags `@incollection` die Daten, die vom Elterneintrag `@collection` geerbt werden, durch einen kurzen Zeiger auf den entsprechenden Elterneintrag ersetzt.

[1] Audrey Author: *Title of article*. In: [2], pp. 134–165.

[2] Edward Editor, ed.: *Title of collection*. Publisher: Location, 1995.

Eine Möglichkeit um solche Überkreuz-Referenzen zu implementieren besteht darin, sich diese als Zitierungen vorstellen, die den Wert des `xref`- oder `crossref`-Feldes als Eintragungsschlüssel nutzen. Hier ein Beispiel:

```

\ProvidesFile{example.bbx}[2007/06/09 v1.0 biblatex bibliography style]

\DeclareCiteCommand{\bbx@xref} {} {...}% code for cross-references {} {}

\DeclareBibliographyDriver{incollection}{%
... \iffieldundef{xref} {...}% code if no cross-reference
{\bbx@xref{\thefield{xref}}}%
... }

```

Beim Definieren von `\bbx@xref` wurden die Parameter `<precode>`, `<postcode>`, und `<sepcode>` im obigen Beispiel leer gelassen, da sie ohnehin nicht genutzt werden. Die Überkreuz-Referenz wird vom `<loopcode>` von `\bbx@xref` ausgegeben. Für mehr Details über das Feld `xref`, siehe § 2.2.3 und schauen Sie in die Hinweise von § 2.4.1. Werfen Sie außerdem einen Blick auf die Tests von `\iffieldxref`, `\iflistxref`, und `\ifnamexref` in § 4.6.2. Das obige Beispiel könnte auch mit Befehl `\entrydata` aus § 4.4.1 implementiert werden:

```

\ProvidesFile{example.bbx}[2007/06/09 v1.0 biblatex bibliography style]

\DeclareBibliographyDriver{incollection}{%
...
\iffieldundef{xref} {...}% code if no cross-reference
{\entrydata{\thefield{xref}}}{% % code for cross-references
... }}%
... }

```

#### 4.11.7. Nutzung eines Interpunktions-Trackers

**4.11.7.1. Die Grundlagen** Es gibt grundlegendes Prinzip, welches Stil-Autoren stets beachten sollten, wenn sie einen bibliographischen Treiber entwickeln: Block- und Unit-Interpunktion werden asynchron gehandhabt. Am besten lässt sich dies an einem Beispiel erklären. Betrachten Sie den folgenden Code-Ausschnitt:

```

\printfield{title}%
\newunit
\printfield{edition}%
\newunit
\printfield{note}%

```

Gibt es kein `edition`-Feld, so wird nicht dieser Code ausgegeben:

```
Title. . Note
```

sondern dieser:

Title. Note

da der Tracker der Unit-Interpunktion asynchron arbeitet. `\newunit` gibt die Unit-Interpunktion nicht sofort aus. Es hält lediglich die Bindung an eine Unit fest und legt `\newunitpunct` auf dem Interpunktionspuffer ab. Dieser Puffer wird vom *nachfolgenden* `\printfield`, `\printlist`, oder ähnlichen Befehlen gehandhabt, aber nur dann, wenn das entsprechende Feld oder die Liste definiert ist. Befehle wie `\printfield` betrachten vorher drei Faktoren, bevor eine Block- oder Unit-Interpunktion eingefügt wird:

- Wurde ein neuer Block oder eine neue Unit angefordert?  
= Ist ein `\newunit`- oder `\newblock`-Befehl vorangegangen?
- Gab das vorangegangene Kommando etwas aus?  
= Ist ein `\printfield` oder ein ähnlicher Befehl vorangegangen?  
= Hat dieses Kommando etwas ausgegeben?
- Geben wir nun etwas aus?  
= Ist das Feld oder die Liste, die jetzt verarbeitet wird, definiert?

Block- und Unit-Interpunktionen werden nur dann eingefügt, wenn *alle* diese Bedingungen zutreffen. Betrachten wir noch einmal das obige Beispiel:

```
\printfield{title}%
\newunit
\printfield{edition}%
\newunit
\printfield{note}%
```

Hier sehen Sie, was passiert, wenn das Feld `edition` nicht definiert ist. Das erste `\printfield`-Kommando druckt den Titel und setzt eine interne ‚neuer Text‘-Markierung. Das erste `\newunit` setzt eine interne ‚neue Unit‘-Markierung. Es wurde an dieser Stelle keine Interpunktion gedruckt. Das zweite `\printfield` tut nichts, da das Feld `edition` nicht definiert ist. Das nächste `\newunit` setzt erneut einen internen Marker ‚neue Unit‘. Noch immer wird keine Interpunktion gedruckt. Das dritte `\printfield` überprüft, ob das Feld `note` definiert ist. Ist dies der Fall, so schaut es auf die ‚neuer Text‘- und ‚neue Unit‘-Marken. Sind beide gesetzt, fügt es den Interpunktionspuffer ein, bevor es die Notiz druckt. Danach löscht es die Marke ‚neue Unit‘ und setzt die ‚neuer Text‘-Marke wieder.

Dies klingt alles komplizierter, als es in Wirklichkeit ist. Für die Praxis bedeutet dies, dass es möglich ist, große Teile eines bibliographischen Treiber am Stück

zu schreiben. Der Vorteil dieses Ansatzes wird klar, wenn Sie versuchen, den obigen Code ohne Interpunktionsstracker zu schreiben. Solch ein Versuch wird in einer Menge von `\iffieldundef`-Tests enden, die nötig sind, um alle möglichen Feldkombinationen zu überprüfen. Bedenken Sie, dass das unten stehende Beispiel nur drei Felder nutzt. Ein gewöhnlicher Treiber benötigt nicht selten zwanzig Felder:

```
\iffieldundef{title}%
{\iffieldundef{edition}
{\printfield{note}} {\printfield{edition}%
\iffieldundef{note}%
{} {.
\printfield{note}}}} {\printfield{title}%
\iffieldundef{edition} {} {.
\printfield{edition}}}%
\iffieldundef{note} {} {.
\printfield{note}}}%
```

**4.11.7.2. Häufige Fehlerquellen** Eine häufige Fehlerquelle gründet darauf, dass man sich die Unit-Interpunktionen als etwas vorstellt, das synchron gehandhabt wird. Dies führt oft zu Problemen, wenn der Treiber wortwörtlichen Text enthält. Betrachten Sie diesen fehlerhaften Code-Ausschnitt, der eine fehl platzierte Unit-Interpunktion generiert:

```
\printfield{title}%
\newunit
(\printfield{series}
\printfield{number})%
```

Dieser Code führt zur folgendem Resultat:

```
Title (. Series Number)
```

Das erste `\printfield` druckt den Titel. Dann markiert `\newunit` eine Unit-Bindung, gibt aber nichts aus. Die Unit-Interpunktion wird vom *nächsten* `\printfield` gedruckt. Dies ist der asynchrone Teil, der zuvor erwähnt wurde. Die öffnende runde Klammer wird direkt gedruckt, bevor in das nächste `\printfield` die Unit-Interpunktion eingefügt wird. Dies führt zur Fehlplatzierung. Wird ein wortwörtlicher Text oder ein Klammerpaar eingefügt (dazu zählen auch diese, die von Befehlen wie `\bibopenparen` und `\mkbibparens` gedruckt werden), so tun sie dies immer in einem `\printtext`-Kommando. Damit der Interpunktionsstracker wie erwartet arbeitet, muss er von allen wortwörtlichen Texten wissen, die vom Treiber eingefügt werden. Das ist der Sinn von `\printtext`. `\printtext` arbeitet mit dem Interpunktionsstracker und stellt sicher, dass der Interpunktionspuffer eingefügt wird,

bevor der wortwörtliche Text gedruckt wird. Es setzt außerdem die interne Marke ‚neuer Text‘. Beachten Sie, dass in diesem Beispiel eine drittes Stück wortwörtlicher Text vorkommt: der Leerraum nach `\printfield{series}`. Im korrigierten Beispiel nutzen wir den Interpunktionsstracker, um diesen Leerraum zu behandeln.

```
\printfield{title}%
\newunit
\printtext{()%
\printfield{series}%
\setunit*{\addspace}%
\printfield{number}%
\printtext{}}%
```

Obwohl der obige Code wie gewünscht arbeitet, wird es empfohlen, Klammern, Anführungszeichen und andere Dinge, die mehr als ein Feld umgeben, durch das Definieren eines Feldformates zu behandeln:

```
\DeclareFieldFormat{parens}{\mkbibparens{#1}}
```

Feldformate können sowohl mit `\printfield`, als auch mit `\printtext` genutzt werden. Deshalb können wir sie nutzen, um mehrere Felder mit einem einzigen Klammerpaar zu umgeben:

```
\printtext[parens]{%
\printfield{series}%
\setunit*{\addspace}%
\printfield{number}%
}%
```

Wir müssen uns noch immer um die Fälle kümmern, in denen keine Information zur Reihe vorliegen:

```
\iffieldundef{series} {}
{\printtext[parens]{%
\printfield{series}%
\setunit*{\addspace}%
\printfield{number}}}%
```

Ein letzter Hinweis: Lokalisierungs-Strings sind kein wörtlicher Text, wenn es um den Interpunktionsstracker geht. Da `\bibstring` und ähnliche Kommandos mit dem Interpunktionsstracker zusammenarbeiten, ist es nicht nötig, sie in einen `\printtext`-Befehl zu schreiben.

**4.11.7.3. Fortgeschrittene Nutzung** Der Interpunktionstracker kann auch genutzt werden, um komplexere Szenarien zu behandeln. Nehmen sie z. B. an, wir wollen die Felder `location`, `publisher` und `year` in einem der folgenden Formate wiedergeben, abhängig von den verfügbaren Daten:

```
...text.
Location: Publisher, Year. Text... ...text.
Location: Publisher. Text... ...text.
Location: Year. Text... ...text.
Publisher, Year. Text... ...text.
Location. Text... ...text.
Publisher. Text... ...text.
Year. Text...
```

Dieses Problem kann (komplizierter Weise) mit einer Menge `\iflistundef`- und `\iffieldundef`-Tests gelöst werden, die alle möglichen Feldkombinationen überprüfen:

```
\iflistundef{location}
{\iflistundef{publisher} {\printfield{year}} {\printlist{publisher}%
\iffieldundef{year} {} {, \printfield{year}}}} {\printlist{location}%
\iflistundef{publisher}%
{\iffieldundef{year} {} {: \printfield{year}} } {: \printlist{publisher}%
\iffieldundef{year} {} {, \printfield{year}}}}}%
```

Das obige Beispiel könnte leserlicher geschrieben werden, indem `\ifthenelse` und die boolischen Operatoren genutzt werden, die in § 4.6.3 besprochen wurden. Dieser Ansatz würde im Grunde der Gleiche sein. Jedoch kann man es auch der Reihe nach schreiben:

```
\newunit \printlist{location}%
\setunit*{\addcolon\space}%
\printlist{publisher}%
\setunit*{\addcomma\space}%
\printfield{year}%
\newunit
```

In der Praxis werden Sie oft eine Kombination aus expliziten Tests und impliziten Tests (die vom Interpunktionstracker durchgeführt werden) anwenden. Beispielsweise, betrachten Sie das folgende Format (beachten Sie die Interpunktion nach dem Ort, wenn es keinen Verlag gibt):

```
...text.
Location: Publisher, Year. Text... ...text.
Location: Publisher. Text... ...text.
Location, Year. Text... ...text.
```

```
Publisher, Year. Text... ..text.
Location. Text... ..text.
Publisher. Text... ..text.
Year. Text...
```

Dies kann durch den folgenden Code gehandhabt werden:

```
\newunit \printlist{location}%
\iflistundef{publisher} {\setunit*{\addcomma\space}}
{\setunit*{\addcolon\space}}% \printlist{publisher}%
\setunit*{\addcomma\space}%
\printfield{year}%
\newunit
```

Da die Interpunktion nach dem Ort eine andere ist, fall es keinen Verlag gibt, benötigen wir einen `\iflistundef`-Test, um diesen Fall abzufangen. Alles anderen wird vom Interpunktionstracker behandelt.

#### 4.11.8. Eigene Lokalisierungsmodule

Stil-Guides enthalten Vorschriften, beispielsweise wie Strings, wie z. B. ‚edition‘ abgekürzt werden sollten, oder sie ordnen Ihnen an, bestimmte Ausdrücke zu nutzen. So zwingt beispielsweise der Stil-Guide `m1a` den Autor dazu, den Term ‚Works Cited‘ anstelle von ‚Bibliography‘ oder ‚References‘ in der Überschrift der Bibliographie zu nutzen. Lokalisierungsbefehle, wie `\DefineBibliographyStrings` aus § 3.7, können in `cbx`- und `bbx`-Dateien genutzt werden, um solche Fälle zu handhaben. Jedoch wird das Überladen von Stil-Dateien mit Übersetzungen nicht empfohlen. Hier kommt `\DeclareLanguageMapping` aus § 4.9.1 ins Spiel. Dieses Kommando bildet eine `l1bx`-Datei mit einer alternativen Übersetzung auf eine Sprache von `babel` ab. Zum Beispiel könnten Sie eine Datei `names french-humanities.l1bx` erstellen, welche französische Übersetzungen für die Nutzung in Geisteswissenschaften bereitstellt und diese auf die `babel`-Sprache `french` abbilden. Dies erfolgt in der Präambel oder in der Konfigurationsdatei:

```
\DeclareLanguageMapping{french}{french-humanities}
```

Wenn die Sprache des Dokuments auf `french` gesetzt wird, wird `french-humanities.l1bx` gegen `french.l1bx` ersetzt. Um auf das Beispiel mit `m1a` zurückzukommen: Ein `m1a`-Stil kann mit einer `american-m1a.l1bx`-Datei daherkommen, um Strings zu liefern, welche dem Stil-Guide `m1a` entgegenkommen. Die Datei würde die folgende Abbildung in der `cbx` und/oder `bbx`-Datei angeben:

```
\DeclareLanguageMapping{american}{american-m1a}
```

Da die alternative l<sub>bx</sub>-Datei Strings vom Standardmodul `american.lbx` erben kann, kann `american-mla.lbx` wie folgt geschrieben werden:

```
\ProvidesFile{american-mla.lbx}[2008/10/01 v1.0 biblatex localization]
\InheritBibliographyExtras{american}
\DeclareBibliographyStrings{%
inherit = {american},
bibliography = {{Works Cited}{Works Cited}},
references = {{Works Cited}{Works Cited}},
}
\endinput
```

Die l<sub>bx</sub>-Dateien müssen gewährleisten, dass das Lokalisierungsmodul komplett ist. Dies tun sie in der Regel, wenn sie Daten vom zugehörigen Standardmodul erben. Wird die Sprache `american` auf `american-mla.lbx` abgebildet, so wird `biblatex american.lbx` nicht laden, solange dieses Modul nicht explizit angefordert wird. Im obigen Beispiel führt das Erben von ‚strings‘ und ‚extras‘ dazu, dass `biblatex american.lbx` lädt, bevor es die Modifikationen an `american-mla.lbx` anwendet.

Beachten Sie, dass `\DeclareLanguageMapping` nicht dazu gedacht ist verschiedene Sprachvarianten (z.B. amerikanisches und britisches Englisch), oder Sprachalias von `babel` (z.B. `USenglish` vs. `american`) zu handhaben. Beispielsweise bietet `babel` die Option `USenglish`, welche der Option `american` ähnelt. Aus diesem Grund kommt `biblatex` mit einer `USenglish.lbx`-Datei, die einfach alle Daten von `american.lbx` erbt. `american.lbx` bekommt die ‚strings‘ wiederum von `english.lbx`. Anders ausgedrückt, das Abbilden von Sprachvarianten und Sprachaliasen von `babel` geschieht auf der Ebene der Dateien. Somit kann der Sprachsupport von `biblatex` einfach dadurch erweitert werden, dass zusätzliche l<sub>bx</sub>-Dateien hinzugefügt werden. Zentralisiertes Abbilden ist nicht nötig. Wenn Sie z. B. Support für Portugiesisch benötigen (in `babel: portugues`), so können Sie eine Datei mit Namen `portugues.lbx` anlegen. Wenn `babel` einen Alias `brasil` anbietet, so würden Sie die Datei `brasil.lbx` anlegen und sie die Daten von `portugues.lbx` erben lassen. Im Gegensatz dazu liegt der Nutzen von `\DeclareLanguageMapping` darin, mit *stilistischen* Sprachvarianten wie ‚Geisteswissenschaften vs. Naturwissenschaften‘, oder ‚mla vs. apa‘ umzugehen, welche typischer Weise am Anfang einer existierenden l<sub>bx</sub>-Datei festgelegt werden.

#### 4.11.9. Gruppieren

In einem Zitierungs- oder Bibliographiestil werden Sie eventuell Markierungen setzen müssen oder bestimmte Werte speichern, um diese später zu nutzen. Für diesen Fall ist es essentiell, die grundlegende Gruppierungsstruktur dieses Paketes zu verstehen. Als Daumenregel gilt: Wenn Autoren-Kommands (wie die, die in § 4.6

besprochen wurden) verfügbar sind, arbeiten Sie in einer großen Gruppe, da das Autoren-Interface dieses Pakets nur lokal aktiviert ist. Sollten bibliographische Daten vorhanden sein, so gibt es mindestens eine zusätzliche Gruppe. Hier sind einige grundlegende Regeln:

- Die gesamte Referenzliste, die von `\printbibliography` und ähnlichen Befehlen ausgegeben wird, wird in einer Gruppe verarbeitet. Jeder Eintrag in dieser Liste wird in einer zusätzlichen Gruppe verarbeitet, die sowohl den `\item code` von `\defbibenvironment`, als auch den Code für die Treiber umfasst.
- Die gesamte Stenogrammliste, die von `\printshorthands` ausgegeben wird, wird in einer Gruppe verarbeitet. Jeder Eintrag in dieser Liste wird in einer zusätzlichen Gruppe verarbeitet, die sowohl den `\item code` von `\defbibenvironment`, als auch den Code für die Treiber umfasst.
- Alle Zitierungsbefehle, die mit `\DeclareCiteCommand` definiert wurden, werden in einer Gruppe verarbeitet. Diese enthält die `\precode`, `\sepcode`, `\loopcode`, und `\postcode` Parameter. Der `\loopcode` wird bei jeder Ausführung von einer zusätzlichen Gruppe umgeben. Sollte ein `\wrapper`-Code spezifiziert worden sein, so wird die gesamte Einheit aus Wrappercode und Zitierungskode von einer zusätzlichen Gruppen zusammengefasst.
- Zusätzlich zu den Gruppierungen, die durch alle Kommandos entstehen, die mit `\DeclareCiteCommand` definiert wurden, führen alle `\autocite` und `\multicite`-Definitionen zu einer weiteren Gruppe.
- `\printfile`, `\printtext`, `\printfield`, `\printlist` und `\printnames` formieren Gruppen. Dies bedeutet, dass alle Formatierungsbefehle in einer eigenen Gruppe verarbeitet werden.
- Alle `lbx`-Dateien werden in einer Gruppe geladen und verarbeitet. Enthält eine `lbx`-Datei Code, der nicht Teil von `\DeclareBibliographyExtras` ist, so müssen die Definitionen global sein.

Beachten Sie, dass die Nutzung von `\aftergroup` in Zitierungs- und Bibliographiestilen unzuverlässig ist, da sich die genaue Anzahl an Gruppen, welche an einem bestimmten Kontext beteiligt sind, in zukünftigen Versionen dieses Pakets ändern kann. Wenn die obige Liste sagt, dass etwas in einer Gruppe verarbeitet wird, so heißt das, dass es *mindestens eine* Gruppe gibt. Es kann sich auch um mehrfach verschachtelte handeln.

#### 4.11.10. Namensräume

Um das Risiko von Namenskonflikten zu minimieren, werden die Namen von internen Macros meist mit einem kurzen, Paket-spezifischen Präfix versehen. Wenn bei-

spielsweise das Paket `foobar` ein internes Makro benötigt, so würde es üblicherweise `\FB@macro` oder `\foo@macro` genannt werden, anstatt nur `\macro` oder `\@macro`. Hier ist eine Liste von Präfixen, die von `biblatex` genutzt oder empfohlen werden:

- blx** Alle Macros mit Namen wie `\blx@name` sind für die interne Nutzung reserviert. Dies gilt auch für Zählernamen, Längenregister, boolischen Variablen, und so weiter. Diese Macros können so geändert werden, dass sie für ältere Versionen nicht mehr nutzbar sind. Sie können umbenannt oder sogar entfernt werden. Solche Änderungen werden nicht einmal in der Versionsgeschichte oder den Release Notes erwähnt. Kurz gesagt: Nutzen Sie niemals Macros mit `blx` im Namen.
- abx** Macros mit dem Präfix `abx` sind ebenfalls interne Macros, allerdings recht stabil. Es ist zwar vorzuziehen, die Macros vom offiziellen Autoren-Interface zu nutzen, aber in manchen Fällen könnte auch ein Macro mit `abx` sinnvoll sein.
- bbx** Dies ist das empfohlene Präfix für interne Macros, die in Bibliographie-Stilen definiert werden.
- bbx** Dies ist das empfohlene Präfix für interne Macros, die in Zitierungs-Stilen definiert werden.
- lbx** Dies ist das empfohlene Präfix für interne Macros, die in Lokalisierungsmodulen definiert werden. Das Lokalisierungsmodul sollte ein zweites Präfix hinzufügen, welches für die Sprache steht. Beispielsweise würde man ein internes Macros, welches vom spanischen Lokalisierungsmoduls definiert wird, `\lbx@es@macro` nennen.

## Anhang

### A. Standardsetup von crossref

Die folgende Tabelle zeigt die Biber Kreuzreferenz-Regeln, die standardmäßig gelten (siehe §§ 2.4.1 and 4.5.3).

| Types                        |                                                                     | Fields         |                |
|------------------------------|---------------------------------------------------------------------|----------------|----------------|
| Source                       | Target                                                              | Source         | Target         |
| *                            | *                                                                   | crossref       | –              |
|                              |                                                                     | xref           | –              |
|                              |                                                                     | entryset       | –              |
|                              |                                                                     | entrysubtype   | –              |
|                              |                                                                     | execute        | –              |
|                              |                                                                     | label          | –              |
|                              |                                                                     | options        | –              |
|                              |                                                                     | presort        | –              |
|                              |                                                                     | related        | –              |
|                              |                                                                     | relatedstring  | –              |
|                              |                                                                     | relatedtype    | –              |
|                              |                                                                     | shorthand      | –              |
|                              |                                                                     | shorthandintro | –              |
|                              |                                                                     | sortkey        | –              |
| mvbook, book                 | inbook, bookinbook, suppbook                                        | author         | author         |
|                              |                                                                     | author         | bookauthor     |
| mvbook                       | book, inbook, bookinbook, suppbook                                  | title          | maintitle      |
|                              |                                                                     | subtitle       | mainsubtitle   |
|                              |                                                                     | titleaddon     | maintitleaddon |
|                              |                                                                     | shorttitle     | –              |
|                              |                                                                     | sorttitle      | –              |
|                              |                                                                     | indextitle     | –              |
|                              |                                                                     | indexsorttitle | –              |
| mvcollection,<br>mvreference | collection, reference, incollection,<br>inreference, suppcollection | title          | maintitle      |
|                              |                                                                     | subtitle       | mainsubtitle   |
|                              |                                                                     | titleaddon     | maintitleaddon |
|                              |                                                                     | shorttitle     | –              |
|                              |                                                                     | sorttitle      | –              |
|                              |                                                                     | indextitle     | –              |
|                              |                                                                     | indexsorttitle | –              |
| mvproceedings                | proceedings, inproceedings                                          | title          | maintitle      |
|                              |                                                                     | subtitle       | mainsubtitle   |
|                              |                                                                     | titleaddon     | maintitleaddon |
|                              |                                                                     | shorttitle     | –              |
|                              |                                                                     | sorttitle      | –              |
|                              |                                                                     | indextitle     | –              |
|                              |                                                                     | indexsorttitle | –              |

| Types                    |                                              | Fields         |                 |
|--------------------------|----------------------------------------------|----------------|-----------------|
| Source                   | Target                                       | Source         | Target          |
| book                     | inbook, bookinbook, suppbok                  | title          | booktitle       |
|                          |                                              | subtitle       | booksubtitle    |
|                          |                                              | titleaddon     | booktitleaddon  |
|                          |                                              | shorttitle     | -               |
|                          |                                              | sorttitle      | -               |
|                          |                                              | indextitle     | -               |
|                          |                                              | indexsorttitle | -               |
| collection,<br>reference | incollection, inreference,<br>suppcollection | title          | booktitle       |
|                          |                                              | subtitle       | booksubtitle    |
|                          |                                              | titleaddon     | booktitleaddon  |
|                          |                                              | shorttitle     | -               |
|                          |                                              | sorttitle      | -               |
|                          |                                              | indextitle     | -               |
|                          |                                              | indexsorttitle | -               |
| proceedings              | inproceedings                                | title          | booktitle       |
|                          |                                              | subtitle       | booksubtitle    |
|                          |                                              | titleaddon     | booktitleaddon  |
|                          |                                              | shorttitle     | -               |
|                          |                                              | sorttitle      | -               |
|                          |                                              | indextitle     | -               |
|                          |                                              | indexsorttitle | -               |
| periodical               | article, suppperiodical                      | title          | journaltitle    |
|                          |                                              | subtitle       | journalsubtitle |
|                          |                                              | shorttitle     | -               |
|                          |                                              | sorttitle      | -               |
|                          |                                              | indextitle     | -               |
|                          |                                              | indexsorttitle | -               |

## B. Default Sortierungsschemata

### B.1. Alphabetische Schemata 1

Die folgende Tabelle zeigt die standardmäßigen, alphabetischen Sortierungsschemata definiert für den Default-Fall (siehe § 3.4 für Erläuterungen).

| Option | Sorting scheme                                                                                                                                |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| nty    | <pre> presort → sortname → sorttitle → sortyear → volume ↔ mm ↔ author ↔ title ↔ year ↔ 0000 ↔ editor ↔ translator ↔ sorttitle ↔ title </pre> |

| Option | Sorting scheme                                                                                                                                |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| nyt    | <pre> presort → sortname → sortyear → sorttitle → volume ↔ mm ↔ author ↔ year ↔ title ↔ 0000 ↔ editor ↔ translator ↔ sorttitle ↔ title </pre> |
| nyvt   | <pre> presort → sortname → sortyear → volume → sorttitle ↔ mm ↔ author ↔ year ↔ 0000 ↔ title ↔ editor ↔ translator ↔ sorttitle ↔ title </pre> |
| all    | <pre> presort → sortkey ↔ mm </pre>                                                                                                           |

## B.2. Alphabetische Schemata 2

Die folgende Tabelle zeigt die standardmäßigen, alphabetischen Sortierungsschemata für die Standardstile, definiert als Default-Fall (siehe § 3.4 für Erläuterungen).

| Option | Sorting scheme                                                                                                                                             |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| anyt   | <pre> presort → labelalpha → sortname → sortyear → sorttitle → volume ↔ mm ↔ author ↔ year ↔ title ↔ 0000 ↔ editor ↔ translator ↔ sorttitle ↔ title </pre> |
| anyvt  | <pre> presort → labelalpha → sortname → sortyear → volume → sorttitle ↔ mm ↔ author ↔ year ↔ 0000 ↔ title ↔ editor ↔ translator ↔ sorttitle ↔ title </pre> |
| all    | <pre> presort → labelalpha → sortkey ↔ mm </pre>                                                                                                           |

## B.3. Chronologische Schemata

Die folgende Tabelle zeigt die chronologischen Standard-Sortierungsschemata für den Default-Fall definiert (siehe § 3.4 für Erläuterungen).

| Option | Sorting scheme                                                                                                                                                                                                                                                                                                                               |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ynt    | <pre> presort → sortyear      → sortname   → sorttitle         ↪ mm           ↪ year          ↪ author     ↪ title                 ↪ 9999                ↪ editor                                            ↪ translator                                            ↪ sorttitle                                            ↪ title </pre>   |
| ydnt   | <pre> presort → sortyear (desc.) → sortname   → sorttitle         ↪ mm           ↪ year (desc.) ↪ author     ↪ title                 ↪ 9999                ↪ editor                                            ↪ translator                                            ↪ sorttitle                                            ↪ title </pre> |
| all    | <pre> presort → sortkey         ↪ mm </pre>                                                                                                                                                                                                                                                                                                  |

## C. Versionsgeschichte

Diese Versionsgeschichte ist eine Liste von Änderungen, die für den Nutzer des Pakets von Bedeutung sind. Änderungen, die eher technischer Natur sind und für den Nutzer des Pakets nicht relevant sind und das Verhalten des Pakets nicht ändern, werden nicht aufgeführt. Wenn ein Eintrag der Versionsgeschichte ein Feature als *improved* oder *extended* bekannt gibt, so bedeutet dies, dass eine Modifikation die Syntax und das Verhalten des Pakets nicht beeinflusst, oder dass es für ältere Versionen kompatibel ist. Einträge, die als *modified*, *renamed*, oder *removed* deklariert sind, verlangen besondere Aufmerksamkeit. Diese bedeuten, dass eine Modifikation Änderungen in bereits existierenden Stilen oder Dokumenten mit sich zieht. Die Zahlen an der rechten Seite stehen für die relevante Stelle dieser Dokumentation.

### 1.6 2011-07-29

|                                                                           |         |            |
|---------------------------------------------------------------------------|---------|------------|
| Added special field sortshorthand . . . . .                               | 2.2.3   | Biber only |
| Revised options maxnames/minnames . . . . .                               | 3.1.2.1 |            |
| Options maxcitenames/mincitenames now supported by backend . . . . .      | 3.1.2.1 | Biber only |
| Options maxbibnames/minbibnames now supported by backend . . . . .        | 3.1.2.1 | Biber only |
| Added options maxalphanames/minalphanames . . . . .                       | 3.1.2.3 | Biber only |
| Removed local options maxnames/minnames from \printbibliography . . . . . | 3.5.2   |            |
| Removed local options maxitems/minitems from \printbibliography . . . . . | 3.5.2   |            |
| Removed local options maxnames/minnames from \bibbysection . . . . .      | 3.5.2   |            |
| Removed local options maxitems/minitems from \bibbysection . . . . .      | 3.5.2   |            |
| Removed local options maxnames/minnames from \bibbysegment . . . . .      | 3.5.2   |            |

|                                                                         |         |            |
|-------------------------------------------------------------------------|---------|------------|
| Removed local options maxitems/minitems from \bibbysegment . . . . .    | 3.5.2   |            |
| Removed local options maxnames/minnames from \bibbycategory . . . . .   | 3.5.2   |            |
| Removed local options maxitems/minitems from \bibbycategory . . . . .   | 3.5.2   |            |
| Removed local options maxnames/minnames from \printshorthands . . . . . | 3.5.3   |            |
| Removed local options maxitems/minitems from \printshorthands . . . . . | 3.5.3   |            |
| Added special field format volcitevolume . . . . .                      | 3.6.6   |            |
| Added special field format volcitepages . . . . .                       | 3.6.6   |            |
| Added special field hash . . . . .                                      | 4.2.4.1 | Biber only |
| Added \mkcomprange . . . . .                                            | 4.6.4   |            |
| Added \mkfirstpage . . . . .                                            | 4.6.4   |            |
| Removed \mkpagefirst . . . . .                                          | 4.6.4   |            |
| Fixed some bugs                                                         |         |            |

### 1.5a 2011-06-17

Fixed some bugs

### 1.5 2011-06-08

|                                                                    |         |            |
|--------------------------------------------------------------------|---------|------------|
| Added option uniquename=mininit/minfull . . . . .                  | 3.1.2.3 | Biber only |
| Added option uniquelist=minyear . . . . .                          | 3.1.2.3 | Biber only |
| Updated documentation of uniquename counter . . . . .              | 4.6.2   | Biber only |
| Updated documentation of uniquelist counter . . . . .              | 4.6.2   | Biber only |
| Expanded documentation for uniquename/uniquelist options . . . . . | 4.11.4  | Biber only |
| Added editorial role reviser . . . . .                             | 2.3.6   |            |
| Added localization keys reviser, revisers, byreviser . . . . .     | 4.9.2   |            |
| Added bibliography heading none . . . . .                          | 3.5.7   |            |
| Fixed some memoir compatibility issues                             |         |            |

### 1.4c 2011-05-12

Fixed some bugs

### 1.4b 2011-04-12

Fixed some bugs

**1.4a 2011-04-06**

Enable uniquename and uniquelist in all authortitle styles . . . . . 3.3.1

Enable uniquename and uniquelist in all authoryear styles . . . . . 3.3.1

Fixed some bugs

**1.4 2011-03-31**

Added package option uniquelist . . . . . 3.1.2.3 Biber only

Added special counter uniquelist . . . . . 4.6.2 Biber only

Revised and improved package option uniquename . . . . . 3.1.2.3 Biber only

Revised and improved special counter uniquename . . . . . 4.6.2 Biber only

Added `\bibnamedelimi` . . . . . 3.8.1 Biber only

Added `\bibindexnamedelima` . . . . . 3.8.1

Added `\bibindexnamedelimb` . . . . . 3.8.1

Added `\bibindexnamedelimc` . . . . . 3.8.1

Added `\bibindexnamedelimd` . . . . . 3.8.1

Added `\bibindexnamedelimi` . . . . . 3.8.1

Added `\bibindexinitperiod` . . . . . 3.8.1

Added `\bibindexinitdelim` . . . . . 3.8.1

Added `\bibindexinithyphendelim` . . . . . 3.8.1

Fixed conflict with some AMS classes

**1.3a 2011-03-18**

Fixed some bugs

**1.3 2011-03-14**

Support `@thesis` with `isbn` . . . . . 2.1.1

Updated `terseinits` option . . . . . 3.1.2.1

Allow macros in argument to `\addbibresource` and `friends` . . . . . 3.5.1

Allow macros in argument to `\bibliography` . . . . . 3.5.1

Introducing experimental support for Zotero RDF/XML . . . . . 3.5.1 Biber only

Introducing experimental support for EndNote XML . . . . . 3.5.1 Biber only

Added option `citecounter` . . . . . 3.1.2.3

Added `citecounter` . . . . . 4.6.2

Added `\smartcite` and `\Smartcite` . . . . . 3.6.2

|                                                                                                                        |         |            |
|------------------------------------------------------------------------------------------------------------------------|---------|------------|
| Added <code>\smarcites</code> and <code>\Smartcites</code> . . . . .                                                   | 3.6.3   |            |
| Added <code>\svolcite</code> and <code>\Svolcite</code> . . . . .                                                      | 3.6.6   |            |
| Added <code>\bibnamedelima</code> . . . . .                                                                            | 3.8.1   | Biber only |
| Added <code>\bibnamedelimb</code> . . . . .                                                                            | 3.8.1   | Biber only |
| Added <code>\bibnamedelimc</code> . . . . .                                                                            | 3.8.1   |            |
| Added <code>\bibnamedelimd</code> . . . . .                                                                            | 3.8.1   |            |
| Added <code>\bibinitperiod</code> . . . . .                                                                            | 3.8.1   | Biber only |
| Added <code>\bibinitdelim</code> . . . . .                                                                             | 3.8.1   | Biber only |
| Added <code>\bibinithyphendelim</code> . . . . .                                                                       | 3.8.1   | Biber only |
| Expanded documentation . . . . .                                                                                       | 3.11.4  |            |
| Added <code>\langle position \rangle</code> parameter <code>f</code> to <code>\DeclareAutoCiteCommand</code> . . . . . | 4.3.1   |            |
| <b>1.2a 2011-02-13</b>                                                                                                 |         |            |
| Fix in <code>\mkbibmonth</code> . . . . .                                                                              | 4.10.4  |            |
| <b>1.2 2011-02-12</b>                                                                                                  |         |            |
| Added entry type <code>@mvbook</code> . . . . .                                                                        | 2.1.1   |            |
| Added entry type <code>@mvcollection</code> . . . . .                                                                  | 2.1.1   |            |
| Added entry type <code>@mvproceedings</code> . . . . .                                                                 | 2.1.1   |            |
| Added entry type <code>@mvreference</code> . . . . .                                                                   | 2.1.1   |            |
| Introducing remote resources . . . . .                                                                                 | 3.5.1   | Biber only |
| Introducing experimental RIS support . . . . .                                                                         | 3.5.1   | Biber only |
| Added <code>\addbibresource</code> . . . . .                                                                           | 3.5.1   |            |
| <code>\bibliography</code> now deprecated . . . . .                                                                    | 3.5.1   |            |
| <code>\bibliography*</code> replaced by <code>\addglobalbib</code> . . . . .                                           | 3.5.1   |            |
| Added <code>\addsectionbib</code> . . . . .                                                                            | 3.5.1   |            |
| Updated and expanded documentation . . . . .                                                                           | 2.4.1   |            |
| Introducing smart crossref data inheritance . . . . .                                                                  | 2.4.1.2 | Biber only |
| Introducing crossref configuration interface . . . . .                                                                 | 4.5.3   | Biber only |
| Added <code>\DefaultInheritance</code> . . . . .                                                                       | 4.5.3   | Biber only |
| Added <code>\DeclareDataInheritance</code> . . . . .                                                                   | 4.5.3   | Biber only |
| Added <code>\ResetDataInheritance</code> . . . . .                                                                     | 4.5.3   | Biber only |
| Added <code>\ifkeyword</code> . . . . .                                                                                | 4.6.2   |            |

|                                                                |       |
|----------------------------------------------------------------|-------|
| Added \ifentrykeyword . . . . .                                | 4.6.2 |
| Added \ifcategory . . . . .                                    | 4.6.2 |
| Added \ifentrycategory . . . . .                               | 4.6.2 |
| Added \ifdriver . . . . .                                      | 4.6.2 |
| Added \forcsvfield . . . . .                                   | 4.6.4 |
| Extended \mkpageprefix . . . . .                               | 4.6.4 |
| Extended \mkpagetotal . . . . .                                | 4.6.4 |
| Extended \mkpagefirst . . . . .                                | 4.6.4 |
| Added localization key inpreparation . . . . .                 | 4.9.2 |
| Rearranged manual slightly, moving some tables to the appendix |       |

### 1.1b 2011-02-04

|                                                                      |         |            |
|----------------------------------------------------------------------|---------|------------|
| Added option texencoding . . . . .                                   | 3.1.2.1 | Biber only |
| Added option safeinputenc . . . . .                                  | 3.1.2.1 | Biber only |
| Expanded documentation . . . . .                                     | 2.4.3.4 |            |
| Improved mergedate option of bibliography style authoryear . . . . . | 3.3.2   |            |
| Removed pass option of \DeclareSortingScheme . . . . .               | 4.5.1   | Biber only |
| Fixed some bugs                                                      |         |            |

### 1.1a 2011-01-08

|                                                 |       |
|-------------------------------------------------|-------|
| Added unsupported entry type @bibnote . . . . . | 2.1.3 |
| Added \bibliography* . . . . .                  | 3.5.1 |
| Fixed some bugs                                 |       |

### 1.1 2011-01-05

|                                                                            |         |            |
|----------------------------------------------------------------------------|---------|------------|
| Added option maxbibnames . . . . .                                         | 3.1.2.1 |            |
| Added option minbibnames . . . . .                                         | 3.1.2.1 |            |
| Added option maxcitenames . . . . .                                        | 3.1.2.1 |            |
| Added option mincitenames . . . . .                                        | 3.1.2.1 |            |
| Fixed idemtracker=strict and idemtracker=constrict . . . . .               | 3.1.2.3 |            |
| Added option mergedate to bibliography style authoryear . . . . .          | 3.3.2   |            |
| Added support for prefixnumbers to bibliography style alphabetic . . . . . | 3.3.2   |            |
| Made option useprefix settable on a per-type basis . . . . .               | 3.1.3   | Biber only |
| Made option useauthor settable on a per-type basis . . . . .               | 3.1.3   | Biber only |

|                                                                  |         |            |
|------------------------------------------------------------------|---------|------------|
| Made option useeditor settable on a per-type basis . . . . .     | 3.1.3   | Biber only |
| Made option usetranslator settable on a per-type basis . . . . . | 3.1.3   | Biber only |
| Made option skipbib settable on a per-type basis . . . . .       | 3.1.3   | Biber only |
| Made option skiplos settable on a per-type basis . . . . .       | 3.1.3   | Biber only |
| Made option skiplab settable on a per-type basis . . . . .       | 3.1.3   | Biber only |
| Made option dataonly settable on a per-type basis . . . . .      | 3.1.3   | Biber only |
| Made option labelalpha settable on a per-type basis . . . . .    | 3.1.2.3 | Biber only |
| Made option labelnumber settable on a per-type basis . . . . .   | 3.1.2.3 |            |
| Made option labelyear settable on a per-type basis . . . . .     | 3.1.2.3 | Biber only |
| Made option singletitle settable on a per-type basis . . . . .   | 3.1.2.3 | Biber only |
| Made option uniquename settable on a per-type basis . . . . .    | 3.1.2.3 | Biber only |
| Made option indexing settable on a per-type basis . . . . .      | 3.1.2.1 |            |
| Made option indexing settable on a per-entry basis . . . . .     | 3.1.2.1 |            |
| Extended \ExecuteBibliographyOptions . . . . .                   | 3.2.2   |            |
| Added \citedate . . . . .                                        | 3.6.5   |            |
| Improved static entry sets . . . . .                             | 3.10.5  | Biber only |
| Introducing dynamic entry sets . . . . .                         | 3.10.5  | Biber only |
| Added \defbibentryset . . . . .                                  | 3.5.10  | Biber only |
| Added option mcite . . . . .                                     | 3.1.1   | Biber only |
| Added mcite/mciteplus-like commands . . . . .                    | 3.6.10  | Biber only |
| Added \sortalphaothers . . . . .                                 | 3.8.1   | Biber only |
| Extended \DeclareNameFormat . . . . .                            | 4.4.2   |            |
| Extended \DeclareIndexNameFormat . . . . .                       | 4.4.2   |            |
| Extended \DeclareListFormat . . . . .                            | 4.4.2   |            |
| Extended \DeclareIndexListFormat . . . . .                       | 4.4.2   |            |
| Extended \DeclareFieldFormat . . . . .                           | 4.4.2   |            |
| Extended \DeclareIndexFieldFormat . . . . .                      | 4.4.2   |            |
| Added \DeclareNameFormat* . . . . .                              | 4.4.2   |            |
| Added \DeclareIndexNameFormat* . . . . .                         | 4.4.2   |            |
| Added \DeclareListFormat* . . . . .                              | 4.4.2   |            |
| Added \DeclareIndexListFormat* . . . . .                         | 4.4.2   |            |
| Added \DeclareFieldFormat* . . . . .                             | 4.4.2   |            |

|                                                                                             |         |            |
|---------------------------------------------------------------------------------------------|---------|------------|
| Added <code>\DeclareIndexFieldFormat*</code> . . . . .                                      | 4.4.2   |            |
| Introducing configurable sorting schemes                                                    |         | Biber only |
| Added <code>\DeclareSortingScheme</code> . . . . .                                          | 4.5.1   | Biber only |
| Added <code>\DeclarePresort</code> . . . . .                                                | 4.5.1   | Biber only |
| Added <code>\DeclareSortExclusion</code> . . . . .                                          | 4.5.1   | Biber only |
| Added <code>\DeclareLabelname</code> . . . . .                                              | 4.5.2   | Biber only |
| Added <code>\DeclareLabelyear</code> . . . . .                                              | 4.5.2   | Biber only |
| Improved special field <code>labelname</code> . . . . .                                     | 4.2.4   | Biber only |
| Improved special field <code>labelyear</code> . . . . .                                     | 4.2.4   | Biber only |
| Added <code>\entrydata*</code> . . . . .                                                    | 4.4.1   |            |
| Added <code>\RequireBiber</code> . . . . .                                                  | 4.6.4   |            |
| Added option check to <code>\printbibliography</code> . . . . .                             | 3.5.2   |            |
| Added option check to <code>\printshorthands</code> . . . . .                               | 3.5.3   |            |
| Added <code>\defbibcheck</code> . . . . .                                                   | 3.5.9   |            |
| Updated support for Portuguese (José Carlos Santos)                                         |         |            |
| Fixed conflict with <code>titletoc</code>                                                   |         |            |
| Fixed some bugs                                                                             |         |            |
| <b>1.0 2010-11-19</b>                                                                       |         |            |
| First officially stable release                                                             |         |            |
| Renamed option <code>bibencoding=inputenc</code> to <code>bibencoding=auto</code> . . . . . | 3.1.2.1 |            |
| Made <code>bibencoding=auto</code> the package default . . . . .                            | 3.1.2.1 |            |
| Added option <code>backend=bibtexu</code> . . . . .                                         | 3.1.2.1 |            |
| Slightly updated documentation . . . . .                                                    | 2.4.3   |            |
| Updated support for Dutch (Alexander van Loon)                                              |         |            |
| Updated support for Italian (Andrea Marchitelli)                                            |         |            |