

Das bibl_atex Paket

Das Benutzerhandbuch

Programmierbares Bibliografieren und Zitieren

Philipp Lehman
(mit Philip Kime, Audrey Boruvka
und Joseph Wright)¹

Version 3.7
20. Oktober 2017

Inhaltsverzeichnis

Tabellenverzeichnis	1	3.2 Globale Anwenderanpas- sung	70
1 Einleitung	2	3.3 Standardstile	70
1.1 Über BibLaTeX	2	3.4 Verwandte Einträge . .	77
1.2 Lizenz	2	3.5 Sortierungsoptionen . .	79
1.3 Rückmeldungen	2	3.6 Daten-Annotationen .	80
1.4 Danksagungen	2	3.7 Bibliografiebefehle . . .	83
1.5 Voraussetzungen	3	3.8 Zitierfehler	101
		3.9 Lokalisierungsbefehle .	113
2 Anleitung zur Datenbasis	6	3.10 Formatierungsbefehle .	115
2.1 Eingabetypen	6	3.11 Anmerkungen zu Spra- chen	127
2.2 Eingabefelder	14	3.12 Anwendungshinweise .	129
2.3 Verwendungshinweise	33	3.13 Hinweise und Warnun- gen	139
2.4 Hinweise und Warnun- gen	42	3.14 Anwenden des alternati- ven BibTeX-Backends	144
3 Benutzerhandbuch	46	4 Versionsgeschichte	146
3.1 Paketooptionen	46		

Tabellenverzeichnis

1 Biber/BibLaTeX Kompatibili- tätsmatrix	7	5 Verbesserte Datumspezifika- tionen	41
2 Unterstützte Sprachen	28	6 Arbeit der Vereindeutigungs- optionen	65
3 Datumsspezifikationen	39	7 Disambiguationscounter . . .	67
4 EDTF 5.2.2 Unspezifisches Da- tenparsing	40	8 mcite-artige Befehle	113
		9 mcite-artige Syntax	114

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.
Anpassung an die Version 3.7 durch Christine Römer. License: L^AT_EX Project Public License (LPPL).
Aufgrund der häufigen Änderungen in letzter Zeit wurde nur die Übersetzung des Nutzerhand-
buchs aktualisiert. Zum Nachschlagen im Autorenhandbuch muss in die englische Originalversion
geschaut werden.

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, 3.7, 3.8, 3.12.

1.1 Über BibLaTeX

Dieses Paket bietet erweiterte bibliografische Möglichkeiten für die Verwendung von LaTeX zusammen mit BibTeX. Das Paket ist eine komplette Neuimplementierung der bibliografischen Einrichtungen, welche von LaTeX zur Verfügung gestellt werden. Das BibLaTeX-Paket arbeitet mit dem „Backend“(Program) Biber, das standardmäßig verwendet wird, das die Dateien im BibTeX-Format verarbeitet und alle Sortierungen, Etikettierungen (und noch viel mehr) vornimmt. Die Formatierung der Bibliografie wird vollständig von T_EX-Makros gesteuert. Gute Kenntnisse in LaTeX sollten ausreichend sein, um eine neue Bibliografie und Zitatvorlage zu erstellen. Dieses Paket unterstützt auch unterteilte Bibliografien, mehrere Bibliografien innerhalb eines Dokuments und separate Listen von bibliografischen Kürzeln. Bibliografien können in Teile unterteilt und/oder nach Themen segmentiert werden. Genau wie die Bibliografiestile, können alle Zitatbefehle frei definiert werden. Es stehen Merkmale wie anpassbare Sortierung, mehrere Bibliografien mit unterschiedlicher Sortierung, anpassbare Etiketten und dynamische Datenänderung zur Verfügung. Bitte beachten Sie den Abschnitt § 1.5.5 mit Informationen zu der Biber-BibLaTeX-Versionskompatibilität. Bitte beachten Sie die Tabelle 2 für eine Liste der Sprachen, die derzeit von diesem Paket unterstützt werden.

1.2 Lizenz

Copyright © 2006–2011 Philipp Lehman, 2012–2016 Philip Kime, Audrey Boruvka, Joseph Wright. Es ist erlaubt, zu kopieren, zu verteilen und /oder die Software zu modifizieren, unter den Bedingungen der LaTeX Project Public License, Version 1.3.³

1.3 Rückmeldungen

Bitte benutzen Sie die BibLaTeX-Projektseite auf GitHub, um Fehler zu melden und um Featureanfragen zu senden.⁴ Bevor Sie eine Featureanfrage senden, stellen Sie bitte sicher, dass 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 oder bei TeX-LaTeX Stack Exchange stellen.⁵

1.4 Danksagungen

Die Sprachmodule dieses Pakets beinhalten die Beiträge von (in der Reihenfolge der Einreichung): Augusto Ritter Stoffel, Mateus Araújo (Brasilianisch); Sebastià Vila-Marta (Katalanisch); Ivo Pletikosić (Kroatisch); Michal Hoftich (Tschechisch); Jonas Nyrup (Dänisch); Johannes Wilm (Danish/Norwegisch); Alexander van Loon, Pieter

²<http://mirror.ctan.org/macros/latex/contrib/biblatex//doc/examples>

³<http://www.latex-project.org/lppl.txt>

⁴<http://github.com/plk/biblatex>

⁵<http://tex.stackexchange.com/questions/tagged/biblatex>

Belmans, Hendrik Maryns (Niederländisch); Hannu Väisänen, Janne Kujanpää (Finnisch); Denis Bitouzé (Französisch); Apostolos Syropoulos, Prokopis (Griechisch); Baldur Kristinsson (Isländisch); Enrico Gregorio, Andrea Marchitelli (Italienisch); Håkon Malmedal (Norwegisch); Anastasia Kandulina, Yuriy Chernyshov (Polnisch); José Carlos Santos (Portugiesisch); Oleg Domanov (Russisch); Tea Tušar and Bogdan Filipič (Slovenisch); Ignacio Fernández Galván (Spanisch); Per Starbäck, Carl-Gustav Werner, Filip Åsblom (Schwedisch); Martin Vrábel, Dávid Lupták (Slovakisch), Benson Muite (Estisch);

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-Paket 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 mit `pdflatex` zu kompilieren.
- Biber** Biber ist das Backend für BibLaTeX zur Übertragung von Daten aus Quelldateien in den LaTeX-Code. Biber kommt mit TeXLive und ist auch erhältlich in SourceForge.⁶ Biber verwendet die „`btparse C library`“ für die BibTeX-Format-Datei-Analyse, die mit den BibTeX-Parsing-Regeln aber vereinbar sein sollte und auch auf die Korrektur der häufigsten Probleme gerichtet ist. Zu den Einzelheiten siehe die Hilfeseite zum „`Perl Text :: BibTeX Modul`“⁷.
- etoolbox** Dieses LaTeX-Paket, das automatisch geladen wird, bietet generische Programmierung, welche zur Einrichtungen von BibLaTeX erforderlich ist. Es ist über CTAN⁸ erhältlich.
- kvoptions** Dieses LaTeX-Paket, das auch automatisch geladen wird, wird für interne Optionen verwendet. Es ist im „`oberdiek package bundle`“ von CTAN.⁹
- 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.¹⁰
- xstring** Dieses LaTeX-Paket, das auch automatisch geladen wird, stellt erweiterte Stringverarbeitungs makros bereit. Es ist erhältlich über CTAN.¹¹

⁶<http://biblatex-biber.sourceforge.net/>

⁷<http://search.cpan.org/~ambs/Text-BibTeX>

⁸<http://ctan.org/pkg/kvoptions>

⁹<http://www.ctan.org/pkg/kvoptions>

¹⁰<http://ctan.org/pkg/logreq/>

¹¹<http://ctan.org/pkg/xstring/>

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/polyglossia Die `babel` und `polyglossia` Pakete bieten die Kernarchitektur für mehrsprachigen Satz. Wenn Sie in einer anderen Sprache als in amerikanisches Englisch schreiben, wird die Nutzung dieser Pakete empfohlen. Wenn BibLaTeX geladen wird, wird `babel` oder `polyglossia` 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.¹²

xpatch Das `xpatch`-Paket erweitert die „patching Befehle“ von `etoolbox` für BibLaTeX-Bibliografie-Makros, Treiber und Formatierungsanweisungen.¹³

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`-Paketoptionen in § 3.1.2.1 für weitere Details. Bei der Verwendung des `hyperref`-Pakets, wird empfohlen, vorher `biblatex` zu laden.

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

memoir Bei Verwendung der `memoir`-Klasse sind die Standard-Bibliografie-Überschriften so angepasst, dass sie gut mit dem Standard-Layout dieser Klasse zusammenpassen. Siehe § 3.13.2 für weitere Verwendungshinweise.

KOMA-Script Bei Verwendung der `scrartcl`-, `scrbook`- oder `scrreprt`-Klassen, die Standard-Bibliografie-Überschriften werden so angepasst, dass sie gut mit dem Standard-Layout dieser Klassen zusammenpassen. Siehe § 3.13.1 für die weitere Verwendungshinweise.

1.5.4 Inkompatible Pakete

Die in diesem Abschnitt aufgeführten Pakete sind nicht kompatibel mit BibLaTeX. Seit der Reimplementierung der Bibliografieausstattung 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.

¹²<http://ctan.org/pkg/csquotes/>

¹³<http://ctan.org/pkg/xpatch/>

- babelbib** Das `babelbib`-Paket bietet Unterstützung für mehrsprachige Bibliografien. Dies ist ein Standardfeature von BibLaTeX. Verwenden Sie das Feld `langid` und die Paketoption `autolang` für ähnliche Funktionen. Beachten Sie, dass BibLaTeX automatisch die Sprache entsprechend dem Hauptdokument anpasst, wenn `babel` oder `polyglossia` geladen ist. Sie brauchen nur die oben genannten Funktionen, falls Sie die Sprachen auf einer pro-Eintrag-Grundlage innerhalb der Bibliografie ändern wollen. Siehe §§ 2.2.3 und 3.1.2.1 für Details. Sehen Sie auch § 3.9.
- backref** Das `backref`-Paket schafft Rückverweise in die Bibliografie. Siehe die Paketoptionen `hyperref` und `backref` in § 3.1.2.1 für vergleichbare Funktionalität.
- bibtopic** Das `bibtopic`-Paket bietet Unterstützung für Bibliografien gegliedert nach Themen, Typ oder anderen Kriterien. Für Bibliografien, unterteilt nach Themen, finden Sie in der Kategorie Feature (in § 3.7.6) und die entsprechenden Filter in § 3.7.2. Alternativ können Sie das Feld `keywords` in Verbindung mit den `keyword` und `notkeyword` Filter für vergleichbarer Funktionalität verwenden, vgl. §§ 2.2.3 and 3.7.2 für weitere Einzelheiten. Für Bibliografien unterteilt nach Typ, nutzen sie den `type` und `nottype` Filtern. Siehe auch § 3.12.4 für Beispiele.
- bibunits** Das `bibunits`-Paket bietet Unterstützung für mehrere Teilbibliografien (z. B. pro Kapitel). Siehe `chapterbib`.
- chapterbib** Das `chapterbib`-Paket bietet Unterstützung für mehrere Teilbibliografien. 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 §§ 3.7.4, 3.7.5, 3.7.2 für weitere Einzelheiten. Siehe auch § 3.12.3 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 § 3.10.
- citeref** Ein weiteres Paket für die Erstellung für Rückverweise in die Bibliografie. Sehen Sie 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 Bibliografie aufgelistet werden. Die Zitatgruppen sind so definiert, wie die Elemente zitiert wurden. Dies funktioniert nur mit unsortierten Bibliografien. Das Paket `biblatex`

unterstützt auch gruppierte Zitate, welche in diesem Handbuch ‚Eintragstypen/entry sets‘ oder ‚Referenztypen/reference sets‘ heißen. Siehe §§ 3.12.5, 3.7.11, 3.8.10 für Einzelheiten.

- mciteplus** Eine deutlich verbesserte Neuimplementierung des `mcite`-Paketes, die Gruppierung in sortierten Bibliografien unterstützt. Siehe `mcite`.
- multibib** Das `multibib`-Paket bietet Unterstützung für Bibliografien, 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 § 3.8 und die Einstellungen in den §§ 3.7.7, 3.7.8, 3.10 für vergleichbare Funktionalität. Siehe auch § 3.8.9.
- splitbib** Das `splitbib`-Paket bietet Unterstützung für Bibliografien, welche thematisch unterteilt sind. Siehe `bibtopic`.
- titlesec** Das `titlesec`-Paket redefiniert Befehle der „user-level-Document-Division“, solche wie `\chapter` oder `\section`. Dieser Ansatz ist nicht kompatibel mit dem Wechsel interner Befehle, angewendet bei BibLaTeX `srefsection` und `refsegment`-Optioneneinstellungen, beschrieben in § 3.1.2.1.
- 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.

1.5.5 Kompatibilitätsmatrix Biber

Die Biber-Versionen sind eng verbunden mit den BibLaTeX-Versionen. Sie benötigen die richtige Kombination der beiden. Biber wird Sie im Kombilierungsprozess warnen, wenn Informationen auf Inkompatibilitäten mit der BibLaTeX-Version auftreten. Tabelle 1 zeigt eine Kompatibilitätsmatrix für die aktuellen Versionen.

2 Anleitung zur Datenbasis

Dieses Kapitel beschreibt das Standarddatenmodell, welches in der Datei `blx-dm.def`, die Teil von `biblatex` ist, definiert wurde. Das Datenmodell wurde definiert unter Benutzung der Makros, die im Kapitel § 4.5.4 (engl. Gesamtversion) dokumentiert sind. Es ist möglich, das Datenmodell neu zu definieren, das BibLaTeX und Biber verwendet, so kann die Datenquelle neue Eintragstypen und Felder (was Stilunterstützung verlangt) enthalten. Die Spezifikation des Datenmodells ermöglicht auch Beschränkungen, so dass Datenquellen so definiert werden, dass sie gegen das Datenmodell validiert werden (nehmend Biber's `--validate-Datenmodell`-Option). Nutzer, die das Datenmodell anpassen möchten, müssen dafür in die `blx-dm.def` schauen und § 4.5.4 (engl. Gesamtversion) lesen.

2.1 Eingabetypen

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

Biberversion	BibLaTeX version
2.6	3.5
2.5	3.4
2.4	3.3
2.3	3.2
2.2	3.1
2.1	3.0
2.0	3.0
1.9	2.9
1.8	2.8
1.7	2.7
1.6	2.6
1.5	2.5
1.4	2.4
1.3	2.3
1.2	2.1, 2.2
1.1	2.1
1.0	2.0
0.9.9	1.7x
0.9.8	1.7x
0.9.7	1.7x
0.9.6	1.7x
0.9.5	1.6x
0.9.4	1.5x
0.9.3	1.5x
0.9.2	1.4x
0.9.1	1.4x
0.9	1.4x

Tabelle 1: Biber/BibLaTeX Kompatibilitätsmatrix

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 Bibliografestils liegt. Die folgende Liste verfolgt daher zwei Ziele. Sie zeigt die Felder, welche von den Standardstilen 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. Bibliografische 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, ISBNs und andere Spezialfelder wie `gender`. Aber diese Muster sind nur nutzbar mit der „Biber `--validate_datamodel`“-Option. 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. Die Spezifikationen des Standarddatenmodells sehen Sie in der Datei `blx-dm.def`, die kommt mit BibLaTeX für eine vollständige Spezifikation.

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`, `issuesubtitle`, `language`, `origlanguage`, `series`, `volume`, `number`, `eid`, `issue`, `month`, `pages`, `version`, `note`, `issn`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`

book Ein Einzelstück, 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 Eintrags `@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 eines 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. einen einzelnen Aufsatz in einer Sammlung von Essays von demselben Autor), ist diese Art vorgesehen 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 formatieren. Die Standardstile behandeln diese Eingabetypen ähnlich wie `@inbook`.

booklet Eine buchähnliche Arbeit ohne formalen Verlag oder eine 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 gibt 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 den `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, origlanguage, volume, part, edition, volumes, series, number, note, publisher, location, isbn, chapter, pages, addendum, pubstate, doi, eprint, eprintclass, 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 von § 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 Schlüssel-liste 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 zu-

sätzlich 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, vom Eintragstyp `@mvproceedings` Gebrauch zu machen.

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 ein Lexikon und Wörterbuch. 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** Ein 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 § 3.12.5 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`
- xdata** Dieser Eintragstyp ist speziell. `@xdata`-Einträge enthalten Daten, die von anderen Einträgen mit dem Feld `xdata` geerbt werden können. Einträge diesen Typs dienen nur als Datencontainer; sie können nicht in der Bibliografie zitiert oder eingefügt werden. Sehen Sie in § 3.12.6 weitere Einzelheiten.
- custom[a-f]** Benutzerdefinierte Typen für spezielle Bibliografiestile. 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 Bibliografiestile verweisen mit ihren Einträgen auf den Alias und nicht auf den Aliasnamen. Alle unbekanntem 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[a--f] bis @customf, d. h., die Standardbibliografiestile bieten keine Unterstützung für diese Typen. Bei der Verwendung von Standardstilen werden sie als @misc-Einträge behandelt.

- artwork Werke der bildenden Kunst wie Gemälde, Skulpturen und Installationen.
- audio Audio-Aufnahmen, in der Regel auf Audio-CD, DVD, Audio-Kassette oder ähnlichen Medien. Siehe auch @music.
- bibnote Ein spezieller Eingabetyp, der nicht wie andere in bib-Dateien verwendet wird. Er ist vorgesehen für Pakete von Drittanbietern wie notes2bib, welche Anmerkungen zu der Bibliografie 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 Bibliografie, während der @bibnote-Typ für Pakete gedacht ist, die Anmerkungen zu den Bibliografie-Einträgen hinzufügen.
- commentary Kommentare, die einen anderen Status als normale Bücher haben, solche wie Rechtskommentare.
- image Bilder, Gemälde, Fotografien und ähnlichen Medien.
- jurisdiction Gerichtsurteile, gerichtliche Aufnahmen und ähnliche Dinge.
- legislation Gesetze, Gesetzesvorlage, Legislativvorschläge und ähnliche Dinge.
- legal Juristische Dokumente wie Verträge.
- letter Persönliche Korrespondenz wie Briefe, E-Mails, Memoiren, etc.
- movie Bewegte Bilder. Siehe auch @video.
- music Musikalische Aufnahmen. Dies ist eine speziellere Variante von @audio.

- performance** Musik- und Theateraufführungen sowie andere Werke der darstellenden Künste. Dieser Typ bezieht sich speziell auf die Veranstaltung als auf eine Aufnahme, eine Partitur oder eine gedruckte Aufführung.
- review** Kritik zu anderen Arbeiten. Dies ist eine spezielle Variante des `@article`-Types. Die Standardstile behandeln diesen Eintrag wie `@article`.
- software** Computersoftware.
- standard** Nationale und internationale Standards durch ein zuständiges Gremium wie die Internationale Organisation für Standardisierung.
- video** Audiovisuelle Aufzeichnungen, in der Regel auf DVD, VHS-Kassette oder ähnlichen Medien. Siehe auch `@movie`.

2.2 Eingabefelder

Dieser Abschnitt gibt einen Überblick über die Eingabefelder, die vom BibLaTeX-Standarddatenmodell unterstützt werden. Sehen Sie in § 2.2.1 für eine Einführung in die Datentypen, die von diesem Paket unterstützt werden und §§ 2.2.2 and 2.2.3 für die konkrete Felderauflistung.

2.2.1 Datentypen

In Datenquellen wie einer `bib`-Datei werden alle bibliografischen 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 bibliografische 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 and 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`.

Namenslistenfelder werden nach dem `\ifuse*`-Test in dem Standardmodell automatisch erstellt (§ ??). Es wird auch automatisch eine `ifuse*`-Option geschaffen, die die Kennzeichnung und die Namenssortierung steuert (§ 3.1.3.1). Biber unterstützt eine anpassbare Liste der Namensteile, aber aktuell ist diese von den gleichen Elementen bestimmt wie bei BibTeX:

- Family name (Familiennamen) (auch erkennbar als ‚letzter‘ Teil)
- Given name (Vorname) (auch erkennbar als ‚erster‘ Teil)
- Namenpräfix (auch erkennbar als ‚von‘ Teil)
- Namenssuffix (auch erkennbar als ‚Jr‘ Teil)

Die unterstützten Namensteile der Liste werden als eine konstante Liste in dem Standarddatenmodell mit dem Befehl `\DeclareDataModelConstant` definiert (??). Allerdings, es ist nicht ausreichend, einfach in diese Liste andere Namensteile zur Unterstützung hinzuzufügen. In der Regel sind sie hart codiert in den Bibliografietreibern und in die Backendverarbeitung. Diese Konstante wird so viel wie möglich verwendet und ist als Grundlage für künftige Verallgemeinerungen und Erweiterungen vorgesehen.

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 § ?? aufgeführt). Wenn ja, wird die lokalisierte Zeichenfolge dargestellt. Wenn nicht, wird das Element ausgegeben. Ein typisches Beispiel für eine Schlüsselliste ist das Element `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 Bereichsfelder bestehen aus einem oder mehreren Bereichen, in denen alle Gedankenstriche normiert sind durch den Befehl `\bibrangedash` ersetzt werden. Ein Bereich ist etwas, dem gegebenenfalls ein oder mehrere Bindestriche folgen, bei einigen keiner (bspw. 5--7). Eine beliebige Anzahl von aufeinander folgenden Bindestrichen ergibt nun nur einen einzigen Bindestrich. Ein typisches Beispiel für einen Felderbereich ist das Feld `pages`. Beachten Sie auch den `\bibrangessep`-Befehl, der verwendet werden kann, das Trennzeichen zwischen mehreren Bereichen anzupassen. Bereichsfelder werden übersprungen und es wird eine Warnung ausgegeben, wenn sie nicht aus einem oder mehreren Bereichen bestehen. Sie können „unordentliche“ Bereichsfelder mit `\DeclareSourceMap` normalisieren, bevor sie analysiert werden (sehen Sie § 4.5.3 (engl. Version)).

Integer fields beinhalten unformatiert Zahlen, die in Ordnungszahlen oder Zeichenfolgen umgewandelt werden können, falls sie ausgegeben werden. Ein typisches Beispiel ist das Feld `extrayear` oder `volume`. Solche Felder werden als ganze Zahlen sortiert. Bibex unternimmt einen (sehr seriösen) Aufwand, um nichtarabische Darstellungen (römische Ziffern zum Beispiel) auf ganze Zahlen für Sortierzwecke abzubilden.

Datepart fields beinhalten unformatierte ganze Zahlen, die zu Ordnungszahlen oder Strings umgewandelt werden können. Ein typisches Beispiel ist das Feld `month`. Für jedes Feld `X` vom Datentyp `date` im Datenmo-

dell werden Datenteil-Felder automatisch erstellt mit den den folgenden Namen:

```
<datatype>year,                <datatype>endyear,
<datatype>month,              <datatype>endmonth,
<datatype>day, <datatype>endday, <datatype>hour,
<datatype>endhour,            <datatype>minute,
<datatype>endminute,          <datatype>second,
<datatype>endsecond,          <datatype>timezone,
<datatype>endtimezone.
```

Date fields beinhalten eine Datumsangabe im Format `yyyy-mm-ddThh:nn[+-][hh[:nn]Z]-` oder einen Datumsbereich im Format `yyyy-mm-ddThh:nn[+-][hh[:nn]Z]/yyyy-mm-ddThh:nn[+-][hh[:nn]Z]-` und andere Formate, erlaubt durch die EDTF-Stufe 1, sehen Sie § 2.3.8. Datumsfelder sind in der Hinsicht speziell, dass sie analysiert und in ihre Komponenten aufgespalten werden. Die `datepart`-Komponenten (sehen Sie oben) werden automatisch definiert und erfasst, wenn ein Feld vom Datentyp `date` im Datenmodell definiert ist. Ein typisches Beispiel ist `date`.

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

URI fields werden im Verbatim-Modus verarbeitet und enthalten Sonderzeichen. Sie sind auch „URL-escaped“, wenn sie nicht aussehen, wie sie es ohnehin schon tun. Das typische Beispiel eines `uri`-Felds ist `url`.

Separated value fields Eine getrennte Liste von wörtlichen Werten. Beispiele sind die Felder `keywords` und `options`. Der Trenner kann konfiguriert sein mit jedem regulären Perl-Ausdruck über die `xsvsep`-Option, standardmäßig die üblichen BibTeX Kommas (optional von Leerzeichen umgeben).

Pattern fields Wörtliches Felder, die mit einem bestimmten Muster übereinstimmen müssen. Ein Beispiel ist das Feld aus § 2.2.3.

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 § ?? aufgeführt). Wenn ja, wird die lokalisierte Zeichenfolge gedruckt. Wenn ja, wird die lokalisierte Zeichenfolge ausgegeben. Wenn nicht, wird der Wert ausgegeben. Ein typisches Beispiel ist das Feld `type`.

Code fields Unterdrücken TeX-Code.

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 BibLaTeX und dessen Backend verwendet wird. Der BibLaTeX-Datentyp ist rechts neben dem Namen angegeben. Siehe § 2.2.1 für Erläuterungen zu der verschiedenen Datentypen.

Einige Felder werden als ‚Label-Felder‘ markiert; dies meint, dass sie oft als Abkürzungsetiketten beim Drucken von Bibliografielisten, im Sinne von Abschnitt § 3.7.3,

verwendet werden. BibLaTeX erstellt automatisch die Unterstützung von Makros für solche Felder. Sehen Sie § 3.7.3.

abstract field (literal)

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

addendum field (literal)

Verschiedene bibliografische Daten, die am Ende des Eintrags gedruckt werden. Dies ist ähnlich dem `note`-Feld, außer dass es am Ende des Bibliografieeintrages 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 Bibliografie automatisch. Siehe auch `introduction` und `foreword`.

annotation field (literal)

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

annotator list (Name)

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

author list (Name)

Der/Die Autor/en des `title`.

authortype field (Schlüssel)

Die Art des Autors. Dieses Feld wirkt sich auf die Zeichenfolge (sofern vorhanden) aus, um den Autor einzuführen. Wird nicht von den Standardbibliografie-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 Seitennumerierungsschemas 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 Bibliografie 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 Bibliografie-Stile wandeln diese in Ordnungszahlen 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`.

entrysubtype field (literal)

Dieses Feld, das nicht von den Standardstilen verwendet wird, kann für das Spezifizieren eines Sybtyps eines Eintragstyps genommen werden. Dies kann für Bibliografistile, die feinkörnigere Gruppe von Eintragstypen unterstützen, verwendet werden.

eprint field (verbatim)

Die elektronische Kennzeichnung einer Online-Publikation. Diese ist in etwa vergleichbar mit DOI, jedoch spezifisch für ein bestimmtes Archiv, Repositorie, Service oder System. Sehen Sie § 3.12.7 zu weiteren Details. Sehen Sie auch zu den Feldern `eprinttype` und `eprintclass`.

eprintclass field (literal)

Weitere Informationen, die durch `eprinttype` gegeben sind. Diese könnten ein Teil eines Archivs, ein Dateipfad, oder eine Klassifizierung etc. sein. Siehe § 3.12.7 für Details. Siehe auch `eprint` und `eprinttype`.

- eprinttype** field (literal)
- Die Art der eprint-Kennung, d. h., der Name des Archivs, einer Quelle, Services oder des System, zu dem eprint gehört. Siehe § 3.12.7 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 benutzerdefinierten Typen, die in § 2.1.3 aufgelistet sind, nützlich sein. Beachten Sie, dass dieses Feld den schlichten/nüchtern 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 Standardbibliografien 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 Bibliografie automatisch. Siehe auch introduction und afterword.
- holder** list (Name)
- Der/Die Inhaber eines Patents (@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 Kategorien 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 Bibliografie automatisch. Siehe auch `foreword` und `afterword`.

isan field (literal)

Die International Standard Audiovisual Number eines audiovisuellen Werkes. Wird nicht von den Standardbibliografistilen 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 Standardbibliografistilen 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 Standardbibliografistilen 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 Bibliografestil erreicht werden, falls gewünscht. Wird nicht von den Standardbibliografestilen verwendet.

location list (literal)

Der/Die Ort/e der Veröffentlichung, d. h. der Standort des `publisher` oder der `institution`, je nach Art des Eingabetyps. Herkömmlich 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 Schlüsselwort behandelt. Diese Liste kann auch für die benutzerdefinierten 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 *Gesammelte Werke*. 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 Bibliografie-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 Bibliografie angegeben wird. Wird nicht von den Standardbibliografestilen 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 bibliografische Daten, die nicht in ein anderes Feld passen. `note` wird verwendet, um bibliografische 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 Patentrechtanfragen.

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 Standardbibliografiestilen 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 Standardbibliografiestilen 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 Standardbibliografiestilen 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 and 3.13.3.

`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)

name

Label field

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)

name

Label field

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.

<code>shorthand</code>	field (literal)	Label field
	Eine besondere Bezeichnung, die bei der Zitierweise anstelle der üblichen verwendet wird. Wenn definiert, überschreibt es das standardmäßige Label. Sehen Sie auch <code>label</code> .	
<code>shorthandintro</code>	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 <code>shorthandintro</code> -Bereich definiert ist, überschreibt es den Standardsatz. Beachten Sie, dass die alternativen Sätze Abkürzungen enthalten müssen.	
<code>shortjournal</code>	field (literal)	Label field
	Eine kurze Version oder eine Abkürzung von <code>journaltitle</code> . Wird nicht von den Standardbibliografestilen verwendet.	
<code>shortseries</code>	field (literal)	Label field
	Eine kurze Version oder eine Abkürzung von <code>series</code> . Wird nicht von den Standardbibliografestilen verwendet.	
<code>shorttitle</code>	field (literal)	Label field
	Der Titel in einer verkürzten Form. Dieses Feld ist in der Regel nicht in der Bibliografie enthalten. Es ist für Zitierungen im Autor-Titel-Format vorgesehen. Falls vorhanden, verwendet der Autor-Titel-Zitatstil dieses Feld anstatt <code>title</code> .	
<code>subtitle</code>	field (literal)	
	Der Untertitel des Werks.	
<code>title</code>	field (literal)	
	Der Titel des Werks.	
<code>titleaddon</code>	field (literal)	
	Ein Anhang zum <code>title</code> , der in einer anderen Schrift ausgegeben wird.	
<code>translator</code>	list (name)	
	Der/Die Übersetzer von <code>title</code> oder <code>booktitle</code> , je nach Art des Eintrags. Wenn der Übersetzer mit dem <code>editor</code> übereinstimmt, verketteten die Standardstile diese Felder in der Bibliografie automatisch.	
<code>type</code>	field (key)	
	Die Art von <code>manual</code> , <code>patent</code> , <code>report</code> , oder <code>thesis</code> . Dieses Feld kann auch für die benutzerdefinierte Typen aus § 2.1.3 nützlich sein.	
<code>url</code>	field (uri)	
	The URL of an online publication. If it is not URL-escaped (no <code>,%'</code> chars), with Biber, it will be URI-escaped according to RFC 3987, that is, even Unicode chars will be correctly escaped.	

`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 (integer)

Das Nummer eines mehrbändigen Buches oder eine Zeitschrift. Es wird eine ganze Zahl erwartet, nicht unbedingt in arabischen Ziffern, da Biber automatisch von römischen Ziffern oder arabischen Zahlen intern auf ganze Zahlen für Sortierzwecke geht. Siehe auch `part`.

`volumes` field (integer)

Die Gesamtzahl der Bände eines mehrbändigen Werkes. Je nach Art des Eintrags bezieht sich dieses Feld auf `title` oder `maintitle`. Es wird eine ganze Zahl erwartet, nicht unbedingt in arabischen Ziffern, da Biber automatisch von römischen Ziffern oder arabischen Zahlen intern auf ganze Zahlen für Sortierzwecke geht.

`year` field (literal)

Das Jahr der Veröffentlichung. Es ist besser das `date` zu verwenden, da dieses auch mit „plain years“ kompatibel ist. Siehe auch § 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 im Standarddatenmodell.

`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 Bibliografie hinzugefügt, auch wenn er nicht explizit zitiert wurde. Man kann dies über die Kommandozeile steuern.¹⁴ 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,

¹⁴siehe die `mincrossrefs` package option in § 3.1.2.1.

aber ihr `crossref` Feld bleibt undefiniert. Ob der Elterneintrag implizit oder explizit in die Bibliografie aufgenommen wurde, spielt keine Rolle. Siehe auch `xref` in this section as well as § 2.4.1.

`entryset` field (separated values)

Dieses Feld ist speziell für Eintragungssätze. Siehe § 3.12.5 für Details. Dieses Feld wird für die Backend-Verarbeitung gebraucht und erscheint nicht in der `.bbl`.

`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 § ??, außer dass es auf einer pro-Eintrag Grundlage in der `bib`-Datei definierbar ist. Jeder Code dieses Felds wird direkt automatisch nach den Hooks ausgeführt.

`gender` field (Pattern matching one of: `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 Bibliografie- 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 Veröffentlichungen werden jedoch solche Schlüsselbegriffe in der Regel in Deutsch gegeben und sind in diesem Fall geschlechtsspezifisch.

`langid` field (identifier)

Die Sprachen-id des Bibliografieeintrags. Die `Alias-hyphenation` wurde für Rückwärtskompatibilität entwickelt. Der Identifizierer muss ein Sprachename aus den Paketen `babel/polyglossia` sein. Diese Informationen können verwendet werden für das Umschalten der Trennmuster und zu lokalisierten Strings in der Bibliografie. Beachten Sie, dass Ländernamen Kasus-sensitiv sind. Die Sprachen, die derzeit von diesem Paket unterstützt werden, sind aufgeführt in table 2. Beachten Sie, dass `babel` die Kennung `english` als alias für `british` oder `american` behandelt, abhängig von der `babel`-Version. Das `BibLaTeX`-Paket behandelt sie immer alias für `american`. Es ist empfehlenswert, die Sprachkennungen `american` und `british` (`babel`) oder eine spezifische Sprachoption zu verwenden, eine spezifische Sprachvariante (`polyglossia` anzugeben in `langidopts` field), um mögliche Verwirrungen zu vermeiden. Vergleichen Sie `language` in § 2.2.2.

`langidopts` field (literal)

Für `polyglossia`-Verwender: Es ermöglicht pro Eintrag spezifische Sprachoptionen. Der wörtliche Wert für dieses Feld ist anzupassen dem `polyglossia`-Sprachenwechsel, wenn die Paketooption `autolang=langname` genommen wird. Beispielsweise die Felder:

Sprache	Region/Dialekt	Babel Identifiers
Katalanisch	Spanien, Frankreich, Andorra, Italien	catalan
Kroatisch	Kroatien, Bosnien und Herzegovina, Serbien	croatian
Tschechisch	Tschechien	czech
Dänisch	Dänemark	danish
Niederländisch	Niederlande	dutch
Englisch	USA	american, USenglish, english
	Großbritannien	british, UKenglish
	Kanada	canadian
	Australien	australian
	Neuseeland	newzealand
Estisch	Estland	estonian
Finnisch	Finnland	finnish
Französisch	Frankreich, Kanada	french, francais, canadien
Deutsch	Deutschland	german
	Österreich	austrian
Deutsch (neu)	Deutschland	ngerman
	Österreich	naustrian
Griechisch	Griechenland	greek
Italienisch	Italien	italian
Norwegisch	Norwegen	norsk, nynorsk, nynorsk
Polnisch	Polen	polish
Portugiesisch	Brasilien	brazil
	Portugal	portuges, portuges
Russisch	Russland	russian
Slowakisch	Slowakei	slovak
Slowenisch	Slovenien	slovene
Spanisch	Spanien	spanish
Schwedisch	Schweden	swedish

Tabelle 2: Unterstützte Sprachen

```
langid      = {english},
langidopts  = {variant=british},
```

würden einbinden den Bibliografieeintrag in:

```
\english[variant=british]
...
\endenglish
```

ids field (separated list of entrykeys)

Zitierungsschlüssel für „aliase“ für den Hauptzierungsschlüssel. Ein Eintrag kann durch eines seiner Aliase und BibLaTeX zitiert werden, als ob man den priären Zitatschlüssel benutzt hätte. Dies geschieht, um Anwendern zu helfen, die ihre Zitierungsschlüssel ändern, jedoch ältere Dokumente haben, die ältere Schlüssel für denselben Eintrag verwenden. Dieses Feld wird bei der Backendverarbeitung gebraucht und erscheint nicht in .bbl.

indexsorttitle field (literal)

Der Titel wird verwendet, um den Index zu sortieren. Im Gegensatz zu `indextitle`, dieses Feld wird nur zur Sortierung verwendet. Der gedruckte Titel vom Index ist aus `indextitle` oder aus dem Feld `title`. Dieses Feld kann nützlich sein, wenn der Titel Sonderzeichen oder Befehle enthält, die die Sortierung des Index stören. Beachten Sie folgendes Beispiel:

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

Stilautoren sollten beachten, dass BibLaTeX den Wert automatisch kopiert, entweder das `indextitle`- oder das `title`-Feld zu `indexsorttitle`, wenn dieses Feld nicht definiert ist.

keywords field (separated values)

Eine getrennte Liste von Schlüsselwörtern. Diese Schlüsselwörter sind für die Bibliografiefilter bestimmt (sehen Sie §§ 3.7.2 and 3.12.4), sie werden in der Regel nicht gedruckt. Beachten Sie, dass der Standardseparator (Komma) Leerräume um den Separator ignoriert.

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

Eine getrennte Liste von Eitragsoptionen in $\langle key \rangle = \langle value \rangle$ -Notation. Dieses Feld wird verwendet, um Optionen pro Eintragsbasis zu setzen. Sehen Sie für Details § 3.1.3 an. Beachten Sie, dass Zitierungen und Bibliografestile zusätzliche Eintragsoptionen hinzufügen können.

presort field (string)

Ein spezielles Feld, um die Sortierreihenfolge der Bibliografie zu modifizieren. Dieses Feld ist der erste Eintrag für die Sortieroutine, wenn die Bibliografie sortiert

wird, daher kann es verwendet werden, um Einträge in Gruppen zu ordnen. Dies kann nützlich sein, wenn Subbibliografien mit Bibliografiefiltern eingerichtet werden. Bitte sehen Sie für weitere Details § 3.5. Beachten Sie auch § ???. Dieses Feld wird von der Backendverarbeitung benutzt und erscheint nicht in `.bbl`.

`related` field (separated values)

Zitierungsschlüssel für andere Einträge, die eine Beziehung zu diesen Einträgen haben. Diese Beziehungen werden jeweils über das Feld `relatedtype` angegeben. Für weitere Details sehen Sie bitte § 3.4 an.

`relatedoptions` field (separated values)

„Per-type options“, um einen Bezugseintrag vorzunehmen. Beachten Sie, dies setzt nicht die Optionen auf den Bezugseintrag selbst, nur `dataonly` klont, was als Datenquelle von dem Elterneintrag verwendet wird.

`relatedtype` field (identifier)

Ein Identifizierer, der die Art der Beziehung für die Schlüsselliste im Feld `related` auflistet. Der Identifizierer ist eine lokalisierte Bibliografiezeichenfolge, gedruckt vor den Daten der Bezugseintragsliste. Es wird auch verwendet, um Art-spezifische Formatierungsanweisungen und Bibliografiemakros für Bezugseinträge zu erkennen. Für weitere Details sehen Sie bitte § 3.4 an.

`relatedstring` field (literal)

Ein Feld, das verwendet wird, um Bibliografiezeichenfolgen, die in dem Feld `relatedtype` angegeben werden, zu überschreiben. Für weitere Details sehen Sie bitte § 3.4 an.

`sortkey` field (literal)

Ein Feld, um die Sortierungsordnung der Bibliografie zu ändern. Stellen Sie sich dieses Feld als Mastersortierungsschlüssel vor. Falls vorhanden, verwendet BibLaTeX dieses Feld während des Sortierens und ignoriert alles andere, mit der Ausnahme des Feldes `presort`. Für weitere Details sehen Sie bitte § 3.5 an. Dieses Feld wird von der den Backendverarbeitung benötigt und erscheint nicht in `.bbl`.

`sortname` list (name)

Ein Name oder eine Liste von Namen, verwendet, um die Sortierreihenfolge der Bibliografie zu modifizieren. Falls vorhanden, wird dieses Feld anstelle von `author` oder `editor` für die Bibliografiesortierung verwendet. Für weitere Details sehen Sie bitte § 3.5 an. Dieses Feld wird von der den Backendverarbeitung benötigt und erscheint nicht in `.bbl`.

`sortshorthand` field (literal)

Ähnlich mit `sortkey`, aber benutzt für die Abkürzungsliste. Wenn vorhanden, benutzt BibLaTeX dieses Feld anstelle von `shorthand` für die Sortierung der Abkürzungsliste. Dies ist nützlich, wenn das Feld `shorthand` Formatierungsbefehle enthält, solche wie `\emph` oder `\textbf`. Dieses Feld wird von der den Backendverarbeitung benötigt und erscheint nicht in `.bbl`.

`sorttitle` field (literal)

Ein Feld, um die Sortierungsordnung der Bibliografie zu modifizieren. Falls vorhanden, wird dieses Feld anstelle des Feldes `title` bei der Bibliografiesortierung verwendet. Das Feld `sorttitle` kann praktisch sein, wenn man einen Eintrag mit einem Titel hat wie „An Introduction to...“ und möchte diesen in der alphabetischen Sortierung unter ‚I‘ und nicht unter ‚A‘ eingeordnet haben. In diesem Fall könnte „Introduction to...“ in das `sorttitle` eingesetzt werden. Für weitere Details sehen Sie bitte § 3.5 an. Dieses Feld wird von der den Backendverarbeitung benötigt und erscheint nicht in `.bbl`.

`sortyear` field (literal)

Ein Feld, um die Sortierungsordnung der Bibliografie zu modifizieren. Falls vorhanden, wird dieses Feld anstelle des Feldes `year` für die Bibliografiesortierung verwendet. Für weitere Details sehen Sie bitte § 3.5 an. Dieses Feld wird von der den Backendverarbeitung benötigt und erscheint nicht in `.bbl`.

`xdata` field (separated list of entrykeys)

Dieses Feld übernimmt Daten von einem oder mehreren `@xdata`-Einträgen. Von der Konzeption her steht das `xdata`-Feld im Zusammenhang mit `crossref` und `xref`: `crossref` etabliert eine logische Eltern/Kind-Relation und erbt Daten; `xref` etabliert eine, logische Eltern/Kind-Relation ohne Daten zu erben; `xdata` erbt Daten ohne eine Relation aufzubauen. Der Wert von `xdata` kann ein einzelner Eitragsschlüssel oder eine getrennte Schlüsselliste sein. Für weitere Details sehen Sie bitte § 3.12.6 an. Dieses Feld wird von der den Backendverarbeitung benötigt und erscheint nicht in `.bbl`.

`xref` field (entry key)

Dieses Feld hat einen alternativen „cross-referencing“-Mechanismus. Er unterscheidet sich von dem Feld `crossref` dahingehend, dass der Kindeintrag keine Daten von Elterneintrag erbt, die im Feld `xref` spezifiziert sind. Wenn die Anzahl der Kindeinträge auf einen spezifischen Elterneintrag verweist, wird eine bestimmte Schwelle gefunden; der Elterneintrag wird der Bibliografie automatisch hinzugefügt, auch wenn er nicht ausdrücklich zitiert wird. Die Schwelle ist einstellbar mit der `mincrossrefs`-Paketoption aus § 3.1.2.1. Stilautoren sollten beachten, ob oder nicht die `xref`-Felder der Kindeinträge auf der untergeordneten Ebene definiert werden, auf der BibLaTeX-Ebene ist es abhängig von der Verfügbarkeit des Eltern-eintrags. Wenn der Elterneintrag vorhanden ist, werden die Felder `xref` der Kindeinträge definiert werden. Wenn nicht, bleiben ihre `xref`-Felder undefiniert. Ob der Elterneintrag eingefügt wird in die Bibliografie wegen der Schwelle oder explizit, weil er zitiert wurde, spielt keine Rolle. Sehen Sie auch zu dem `crossref`-Feld in diesem Abschnitt an sowie § 2.4.1.

2.2.4 Benutzerdefinierte Felder

Die Felder in diesem Abschnitt sind für spezielle Bibliografiestile bestimmt. Sie werden nicht in den Standardbibliografiestilen verwendet.

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

Benutzerdefinierte Listen für spezielle Bibliografiestile. Wird nicht von den Standardbibliografiestilen verwendet.

`name[a-c]type` field (Schlüssel)

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

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

Benutzerdefinierte Listen für spezielle Bibliografestile. Wird nicht von den Standardbibliografestilen verwendet.

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

Benutzerdefinierte Felder für spezielle Bibliografestile. Wird nicht von den Standardbibliografestilen verwendet.

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

Ähnlich wie die benutzerdefinierten Felder oben, außer dass es sich um wörtlich Felder handelt. Wird nicht von den Standardbibliografestilen 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 Bibliografie- 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 and 3.12.7.

`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 (verbatim)

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 3.12.7.

`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 und 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`. Siehe § 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 Bibliografie überhaupt Verweise auf Kapitel oder Abschnitte enthalten sollte, ist umstritten, da ein Kapitel keine bibliografische 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 Bibliografistile 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 Bibliografie 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 (engl. Version) 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 und 3.5 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 Bibliografie 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 Wörtliche Listen

Die Felder `institution`, `organization`, `publisher` und `location` sind wörtliche Listen im Sinne des § 2.2. Dies gilt auch für `origlocation`, `origpublisher` und die Aliasfelder `address` und `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 Shakespeares 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. Der jeweilige Band hat auch einen Titel, und es gibt immer noch den Haupttitel der gesamten Ausgabe:

```
@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 Bibliografie. Wenn ein Eintrag typischerweise mit dem Gründer zitiert wird (und entsprechend in der Bibliografie sortiert), kommt der Gründer in das Feld `editor` und der aktuelle Editor kommt in eines der `editor...-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 §§ 3.9 and ?? für weitere Einzelheiten.

2.3.7 Veröffentlichungen und Zeitschriftenserien

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 Bücher 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 § ?? für eine Liste der standardmäßig definierten Lokalisierungsschlüssel.

Datumspezifikation	Formatierte Daten (Beispiele)	
	Kurz/12-Stundenformat	Lang/24-Stundenformat
1850	1850	1850
1997/	1997-	1997-
1997/unknown	1997-	1997-
1997/open	1997-	1997-
1967-02	02/1967	Februar 1967
2009-01-31	31/01/2009	31ter Januar 2009
1988/1992	1988-1992	1988-1992
2002-01/2002-02	01/2002-02/2002	Januar 2002-Februar 2002
1995-03-30/1995-04-05	30/03/1995-05/04/1995	30ter März 1995-5ter April 1995
2004-04-05T14:34:00	05/04/2004 2:34 PM	5ter April 2004 14:34:00

Tabelle 3: Datumsspezifikationen

2.3.8 Datumsspezifikationen

Datumfelder, wie die Daten des Standarddatenmodells `date`, `origdate`, `eventdate` und `urldate`, halten sich an EDTF¹⁵ (Extended Date/Time Format / Erweitertes Datum/Zeit-Format) Spezifikationsebenen 0 and 1. EDTF ist eine strengere Teilmenge der etwas chaotischen, erlaubten Formate von ISO8601v2004 und ist besser geeignet für bibliografische Daten. Neben dem EDTF leeren Datumsbereichmarker können Sie einen unbefristeten Zeitraum angeben, indem der Bereichstrenner angegeben und das Enddatum weggelassen wird (z. B. `yyyy/`). Sehen Sie Tabelle 3 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 kein `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 § ?? . Siehe diesen Abschnitt und Tabelle ?? auf Seite ?? für weitere Informationen.

Datumfeldnamen *müssen* mit der Zeichenfolge `,date'` enden, wie bei den Standarddatumfeldern. Denken Sie daran, beim Hinzufügen neuer Datenfelder mit dem Datenmodell (sehen Sie § ??). BibLaTeX wird alle Datumfelder nach dem Lesen des Datummodells überprüfen und wird mit einer Fehlerausgabe enden, wenn es ein Datumfeld findet, das nicht dieser Namenskonvention entspricht.

EDTF unterstützt Daten vor der verbreiteten Zeit (BCE/BC) mittels eines negativen Datumsformats und unterstützt `,ungefähre'` (circa) und unsichere Daten. Solche Datenformate setzen interne Marker, man kann entsprechend lokalisierte Marker (wie `circa` oder `beforecommonera`) testen, dies kann eingesetzt sein. Unterstützt werden auch `,unsichere'` Daten ((EDTF 5.2.2), die automatisch expandieren in die entsprechenden Datenbereiche durch das Feld `<datatype>dateunspecified`, die Details der Granularität von un spezifizierten Daten. Sile können diese Informationen benutzen, um solche Termine entsprechend zu formatieren, aber die Standardstile tun dies nicht. Sehen Sie table 4 auf Seite 40 für die erlaubten EDTF `,un spezifizierten'` Formate, deren Programmiererweiterungen und `<datatype>dateunspecified`-Werte (sehen Sie § 4.2.4.1 (engl. Version)).

¹⁵<https://www.loc.gov/standards/datetime/pre-submission.html>

Datenspezifikation	Erweitertes Angebot	Metainformation
199u	1990/1999	yearindecade
19uu	1900/1999	yearincentury
1999-uu	1999-01/1999-12	monthinyear
1999-01-uu	1999-01-01/1999-01-31	dayinmonth
1999-uu-uu	1999-01-01/1999-12-31	dayinyear

Tabelle 4: EDTF 5.2.2 Unspezifisches Datenparsing

Tabelle 5 zeigt Formate, die entsprechenden Prüfungen und Formatierung. Sehen Sie Daten-meta-informationstest und Lokalisierungsstrings. Sehen Sie auch die Beispieldatei `96-dates.tex`, die komplette Beispiele für die Tests und Lokalisierungsstrings verwendet.

Die Ausgabe von ‚circa‘, Unsicherheit und Zeitinformationen in Standardstilen (oder benutzerdefinierte Stilen brauchen nicht die internen `\mkdaterange*`-Makros) wird durch die Paketooptionen `datecirca`, `dateuncertain`, `dateera` und `dateeraauto` gesteuert (sehen Sie § 3.1.2.1). Sehen Sie Tabelle 5 auf Seite 41 für Beispiele, die diese Optionen übernehmen und alle nutzen.

2.3.9 Monats- und Zeitschriftenausgaben

Das Feld `month` ist ein Integerfeld. Die Bibliografiestile 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:

Datumspezifikation	Formatierte Datums (Beispiele)	
	Output-Format	Output-Format-Stichworte
0000	1 BC	dateera=christian druckt beforechrist localisation
-0876	877 BCE	dateera = secular druckt beforecommonera Localisation-string
-0877/-0866	878 BC-867 BC	nehmen \ifdateera Test und beforechrist Localisation-string
0768	0768 CE	nehmen dateeraauto setzt auf ‚1000‘ und commonera Localisation-string
-0343-02	344-02 BCE	
0343-02-03	343-02-03 CE	mit dateeraauto=400
0343-02-03	343-02-02 CE	mit dateeraauto=400 und julian
1723~	circa 1723	nehmen \ifdatecirca Test
1723?	1723?	nehmen \ifdateuncertain Test
1723?~	circa 1723?	nehmen \ifdateuncertain und \ifdatecirca Tests
2004-22	2004	auch, season gesetzt auf den Localisation-string ‚Sommer‘
2004-24	2004	auch, season gesetzt auf den Localisation-string ‚Winter‘

Tabelle 5: Verbesserte Datumspezifikationen

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

Das `bookpagination`-Feld 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 § 3.13.3 für weitere Verwendungshinweise.

Die Felder `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 § 3.9.

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 § ??.

2.4 Hinweise und Warnungen

Dieser Abschnitt enthält einige zusätzliche Hinweise zu der Datenschnittstellen des Pakets. Er geht auch auf einige häufig auftretende Probleme ein.

2.4.1 Querverweise

Biber-Merkmale verfügen über einen hochgradig anpassbaren Querverweismechanismus mit flexiblen Datenvererbungsregeln. Das Duplizieren bestimmter Felder in den Eltern-Eintrag oder das Hinzufügen von leeren Feldern in den Kind-Eintrag ist nicht mehr erforderlich. Einträge werden auf natürliche 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 von den Eltern wird auf das `booktitle`-Feld des Kindes übertragen, wobei der `subtitle` erhält `booksubtitle`. Das `author` der Eltern erhält `bookauthor` des Kindes und da das Kind ein `author`-Feld nicht erhält, ist es auch zu duplizieren wie der `author` des Kindes. Nach der Datenvererbung sieht der Kindeintrag ähnlich zu dem folgenden aus:

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

Im Anhang (engl. Version) finden Sie eine Liste von standardmäßig eingerichteten Zuordnungsregeln. Beachten Sie, dass dies alles anpassbar ist. Sehen Sie in § 4.5.11 (engl. Version) Biber's cross-referencing-Mechanismus konfiguriert ist. Sehen Sie auch § 2.2.3.

2.4.1.1 Das `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 Bibliografie einfügen. Der Schwellenwert wird durch die `minxrefs`-Paketoption aus § 3.1.2.1 gesteuert. Siehe dazu auch. § 2.2.3.

2.4.2 Sortier- und Kodierungsprobleme

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.¹⁶ Sortierungsüberschneidungen auf der Basis von Unicode Common Locale Data Repository (CLDR) werden hinzugefügt.¹⁷

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 § ??) ohne manuellen Eingriff handhaben. Alles was Sie tun müssen, um lokalisierte Sortierung zu erhalten, ist lokal angeben:

```
\usepackage[sortlocale=de]{biblatex}
```

oder wenn Sie „german“ als Hauptdokumentensprache mit Babel oder Polyglossia nehmen:

```
\usepackage[sortlocale=auto]{biblatex}
```

BibLaTeX wird dann Babel/Polyglossia die Hauptdokumentensprache wie die lokale übergeben, so dass Biber eine geeignete lokalen Standards abbildet. Biber wird nicht versuchen, lokale Informationen aus seiner Umgebung zu erhalten, wie dies Dokumentenverarbeitung mit Abhängigkeit von etwas, das nicht im Dokument ist, macht. Das ist gegen den Geist der Reproduzierbarkeit von TeX. Dies macht auch Sinn, da Babel/Polyglossia in der Tat die entsprechende Dokumentumgebung sind. 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 § 2.4.2.1, wie Eingabe- und Datenkodierungen richtig angegeben werden.

2.4.2.1 Besondere Kodierungen Bei Verwendung einer Nicht-Ascii-Kodierung in der `bib`-Datei, ist es wichtig zu verstehen, was BibLaTeX für Sie tun kann und

¹⁶<http://unicode.org/reports/tr10/>

¹⁷<http://cldr.unicode.org/>

was einen manuellen Eingriff erfordert. Das Paket kümmert sich um die LaTeX-Seite, d. h. es wird sichergestellt, dass die Daten aus der `bbl`-Datei richtig importiert werden, sofern Sie die `bibencoding`-Paketoption richtig eingestellt haben. All dies wird 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:

```
\usepackage{biblatex}
```

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

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

- Latin 9-Kodierung (iso-8859-15) in `tex`- und `bib`-Dateien mit pdfTeX oder traditionell:

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

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

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

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

```
\usepackage[bibencoding=latin1]{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 `š` 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 `§`). Wenn Sie vollständige Unicodeunterstützung benötigen, sollten Sie zu XeTeX oder LuaTeX wechseln.

Typische Fehlerangaben, wenn `inputenc` nicht umgehen kann mit UTF-8 Zeichen, sind:

```
! Package inputenc Error: Unicode char <char> (U+<codepoint>)
(inputenc)                not set up for use with LaTeX.
```

aber auch weniger offensichtliche Dinge wie:

```
! Argument of \UTFviii@three@octets has an extra }.
```

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 Bibliografistile erstellen, können sie mit der anschließenden Anleitung für Autoren (in der englischsprachigen Originalversion) fortfahren.

3.1 Paketooptionen

Alle Paketooptionen werden als `<key>=<value>`-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 Load-time-Optionen

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

```
backend=bibtex,bibtex8,bibtexu,biber                                default:biber
```

Einzelheiten der Datenbasenbackends. Die folgende Backends werden unterstützt.

<code>biber</code>	Biber, das Standardbackend von BibLaTeX, unterstützend Ascii, 8-bit Encodings, UTF-8, on-the-fly-Reencoding, lokale-spezifische Sortierung und viele andere Dinge. Lokale-spezifische Sortierung, Fall-sensitive Sortierung und aufwärts /vorrangige Prozesse werden mit den Optionen <code>sortlocale</code> , <code>sortcase</code> and <code>sortupper</code> gesteuert.
<code>bibtex</code>	Eerbtes BibTeX. Traditionelles BibTeX unterstützt nur Ascii-Encoding. Die Sortierung ist immer Fall-insensitive.
<code>bibtex8</code>	<code>bibtex8</code> , die 8-bit-Implementation von BibTeX, unterstützt Ascii- und 8-bit-Encodings, wie beispielsweise Latin 1.

Sehen Sie in § 3.14 nach Details für das Nehmen von BibTeX als Backend.

`style=<file>` default: numeric

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

`bibstyle=<file>` default: numeric

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

`citestyle=<file>` default: numeric

Lädt die Literaturverweisregel `file.cbx`. Einen Überblick zu den Standard-Literaturverweisregel 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 § 3.8.9.

`mcite=true, false` default: false

Lädt ein Literaturverweisregelmodul, das Befehle zur Verfügung stellt, die ähnlich wie bei `mcite/mciteplus` funktionieren. Für Details siehe § 3.8.10.

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 Bibliografie- 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,` default: nty
`<name>`

Die Sortierung in der Bibliografie: 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.

`debug` Sortiert nach dem Eintragungsschlüssel. Nur für die Fehlersuche bestimmt.

`<name>` Benutze `<name>` wie in `\DeclareSortingScheme` definiert (§ 4.5.6; engl. Version)

Egal welches ‚alphabetisches‘ Sortierschema man benutzt, es macht nur Sinn in Verbindung mit einem Bibliografiestil, der die entsprechenden Etiketten ausgibt. Beachten Sie, dass einige Bibliografiestile diese Paketooptionen 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 sowie zum Thema Sortierprozess, schauen sie bitte in § 3.5. Um vordefinierte Schemen zu adaptieren oder neue zu definieren, schauen Sie in § 4.5.6 (engl. Version).

`sortcase=true, false` default: true

Einerlei ob die Bibliografie zu sortieren ist und die Kürzelliste „case-sensitively“ ist.

`sortupper=true, false` default: true

Diese Option entspricht dem Biber-Kommandozeilen-Befehl `--sortupper`. Ist die Option aktiviert, wird die Bibliografie nach ‚Großschreibung vor Kleinschreibung‘ sortiert. Deaktiviert, bedeutet genau anders herum.

`sortlocale=auto, <locale>`

Biber only

Diese Option setzt die lokale Sortierung auf lokal. Jedes Sortierungsschema erbt dies lokal, wenn es keine Spezifizierung der `<locale>`-Option für `\printbibliography` gibt. Fordert diese `auto`, dass für das Hauptdokument die Babel/Polyglossia-Sprachenerkennung eingestellt wird, dann nimmt das Paket diese und ansonsten `en_US`. Biber wird die Babel/Polyglossia-Sprachidentifikatoren in vernünftige lokale Identifikatoren abbilden (sehen Sie dazu die Biber-Dokumentation). Sie können daher die Spezifikation entweder mit einem üblichen lokalen Identifikator wie `de_DE_phonebook`, `es_ES` oder mit dem unterstützten Babel/Polyglossia-Sprachidentifikatoren vornehmen, wenn es für Sie in Ordnung ist, dass Biber dies tut.

`related=true, false` default: true

Unabhängig davon, ob Informationen von verbundenen Einträgen zu nehmen sind oder nicht. Sehen Sie weiter in § 3.4.

`sortcites=true, false` default: false

Unabhängig davon, ob Zitate zu sortieren sind oder nicht, wenn mehrere Eintragungsschlüssel zu einem Zitierbefehl passen. Ist diese Option aktiviert, werden die Literaturverweise nach nach der Reihenfolge der Bibliografie sortiert (sehen Sie § 3.7.10). 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. `maxnames` ist die Masteroption, die sowohl `maxbibnames` als auch `maxcitenames` setzt.

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

Eine Begrenzung, die alle Namenslisten (`author`, `editor`, usw.) beeinflusst. Sollte eine Liste mehr als $\langle maxnames \rangle$ beinhalten, wird sie automatisch auf $\langle minnames \rangle$. Der $\langle minnames \rangle$ -Wert muss natürlich kleiner oder gleich $\langle maxnames \rangle$ sein. `minnames` ist die Masteroption, die sowohl `minbibnames` als auch `mincitenames` setzt.

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

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

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

Ähnlich wie `minnames`, jedoch nur die Bibliografie 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 § 3.8.4 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 Optionen 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 § 3.8 und `\DeclareAutoPunctuation` in § ??.

`language`=`autobib`, `autocite`, `auto`, $\langle language \rangle$ default: `autobib`

Diese Option kontrolliert die Unterstützung von Mehrsprachigkeit. Auf `autobib`, `autocite` oder `auto` gestellt, wird BibLaTeX versuchen, die Hauptsprache des Dokuments aus `babel/polyglossia` zu benutzen (und automatisch Englisch nehmen, sollte `babel/polyglossia` nicht vorhanden sein). Es ist auch möglich, die Dokumentsprache manuell auszuwählen. In diesem Fall wird `langid`

die Einträge überschreiben und Sie sollten noch eine Umgebung für die Sprachumschaltung mit der Option `autolang` wählen, um festzulegen, wie der Schalter auf die manuell gewählte Sprache umgangen wird. Bitte beachten Sie Tabelle 7 für eine Liste der unterstützten Sprachen und ihren entsprechenden Identifikator. `autobib` schaltet die Sprache für jeden Eintrag in der Bibliografie an, das Nehmen des Feldes `langid` und der Sprachenumgebung spezifiziert die `autolang`-Option. `autocite` schaltet die Sprache für jede Zitierung an, durch das Nehmen des `langid`-Felds und der Sprachenumgebung. `auto` ist eine Abkürzung, um sowohl `autobib` und `autocite` zu setzen.

`clearlang=true, false` default: true

Wenn diese Option aktiviert ist, wird BibLaTeX 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 (engl. Version) zur Verfügung gestellt.

`autolang=none, hyphen, other, other*, langname` default: none

Diese Option kontrolliert die `babel`-Sprachumgebung¹⁸, wenn das `babel/polyglossia`-Paket geladen ist und ein Bibliografieeintrag enthält ein `langid`-Feld (beachten Sie § 2.2.3). Berücksichtigen Sie, BibLaTeX asst sich automatisch der Dokumentsprache an, wenn `babel/polyglossia` geladen wurde. In mehrsprachigen Dokumenten wird es sich auch kontinuierlich anpassen, soweit es Zitate und die Bibliografiestandardsprache der aktuellen Sprache betrifft. Die Wirkung der Spracheinstellung hängt von der Sprachumgebung ab, die durch diese Option ausgewählt wird. Die Wahlmöglichkeiten sind:

- `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/polyglossia` und BibLaTeX für die entsprechende Sprache zur Verfügung stellen, werden aktiviert, und Schlüsselemente 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.
- `langname` `polyglossia` nur. Setzen Sie den Eintrag in eine Umgebung `,language\name`. Der Vorteil dieses Optionenwertes für `polyglossia`-Nutzer ist, dass dieser Kenntnis nimmt vom `langidopts`-Feld, sodass sie pro Sprache Optionen in den Eintrag hinzufügen können (wie eine Sprachvariante auszuwählen). Wenn

¹⁸`polyglossia` verwendet die `babel`-Sprachumgebungen auch und so verwendet diese Option beide, die `babel`- und die `polyglossia`-Sprachumgebungen.

`babel` genommen wird, hat diese Option den selben Wert wie der Optionenwert `other`.

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

Diese Option kontrolliert den extra Raum zwischen den einzelnen Einträgen, d. h. die Abstände zwischen den Bibliografieblöcken. 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-Dokument-Klassen.
- `par` Beginne einen neuen Paragraphen für jeden Eintrag. Das hat einen ähnlichen Effekt wie die `openbib`-Option des Standardverhaltens der LaTeX-Dokument-Klassen.
- `nbpar` Ähnlich der `par`-Option, aber verbietet Seitensprünge zwischen den Übergängen und innerhalb der Einträge.
- `ragged` Fügt einen kleinen Strafraum ein, um Zeilenumbrüche an Blockgrenzen zu fördern und die Bibliografie rechts orientiert (Flattersatz) zu setzen.

Der Befehl `\newblockpunct` kann auch verändert werden, um verschiedene Resultate zu erzielen, sehen Sie § 3.10.1. Beachten Sie auch § 4.7.1 (engl. Version) für weitere Informationen dazu.

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

Diese Option kontrolliert das Verhalten von `\mkbibfootnote`, `\mkbibendnote` und ähnliche Wrapper aus § 4.10.4 (engl. Version) 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 Bibliografie geschrieben werden sollen. Die Endreferenzen sind eine Liste von Seitenzahlen, die angeben, auf welchen Seiten der entsprechende Bibliografieeintrag 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 Bibliografiestil diese Möglichkeit unterstützen. Alle Standardstile, die mit diesem Paket versendet werden, tun dies.

`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 sqq., 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 sqq., 21 sqq.‘ 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, setplusem` 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 gesetzten 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 Bibliografie. Genauer gesagt, beeinflusst sie der `\ifciteindex`- und `\ifbibindex`-Befehl aus § 4.6.2 (engl. Version) Die Option kann für alles, eine Pro-Typ- oder eine Pro-Eintrag-Basis gesetzt werden. Dabei stehen folgende Möglichkeiten zur Verfügung:

- `true` Aktiviert Katalogisierung für alles.
- `false` Deaktiviert Katalogisierung für alles.
- `cite` Aktiviert Katalogisierung nur für Literaturverweise.
- `bib` Aktiviert Katalogisierung nur für die Bibliografie.

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 Bibliografie. 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 § 3.12.8 und `\printfile` in § 4.4.1 (engl. Version) 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 § 3.7.4. Dabei stehen folgende Möglichkeiten für die Dokumentenaufspaltung zur Verfügung:

- `none` Deaktiviert dieses Feature.
- `part` Beginnt eine Referenzsektion an jedem `\part` Befehl.
- `chapter` Beginnt eine Referenzsektion an jedem `\chapter` Befehl.
- `section` Beginnt eine neue Referenzsektion an jedem `\section` Befehl.
- `subsection` Beginnt eine neue Referenzsektion an jedem `\subsection` 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 § 3.7.5 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 Referenzsektion angewandt werden.

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

Diese Option führt automatisch den `\citereset`-Befehl von § 3.8.8 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 Bibliografie 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:

- `year` Benutzt nur Jahr, beispielsweise:
2010
2010–2012
- `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>edtf</code>	Nimmt das „strict Extended Date/Time Format“ ¹⁹ (yyyy-mm-dd), beispielsweise: 2010-01-01 2010-01-21/2010-01-30
<code>ymd</code>	Ein Jahr-Monat-Tag-Format, das modifiziert sein kann, mit anderen Optionen als das strikte EDTF, beispielsweise: 2010-1-1 2010-1-21/2010-1-30

Beachten Sie, dass das `edtf`-Format erzwungen wird von `dateera = astronomical`, `datezeros = true`, `timezeros = true`, `seconds = true`, `<datatype>time = 24h` und `julian = false`. `ymd` ist ein ETFT-artiges Format, das aber verschiedene Optionen wechseln kann, die die strikte `edtf`-Option nicht gestattet.

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 für Monate und Tage in allen kurzen Formaten können einzeln durch die `datezeroes`-Paketoption kontrolliert werden. Die vorangestellten Nullen für Stunden, Minuten und Sekunden in allen kurzen Formaten können separat mit der Paketoption `timezeros` gesteuert werden. Beim Ausdrucken von `times`, wird das Ausgeben von Sekunden und Zeitzonen über die Optionen `seconds` und `timezones` gesteuert.

Die Optionen `julian` und `gregorianstart` kann man zur Steuerung benutzen, wenn Julian-Kalenderdaten verwendet werden.

`datelabel=year, short, long, terse, comp, ymd, edtf` default: year

Ähnlich der `date`-Option, kontrolliert aber das Format des `date`-Felds, ausgewählt mit `\DeclareLabeldate`.

`<datatype>date=year, short, long, terse, comp, ymd, edtf` default: comp

Ähnlich der `date`-Option, kontrolliert aber das Format des `<datatype>date`-Felds im Datenmodell.

`alldates=year, short, long, terse, comp, edtf`

Setzt die Option für alle Daten des Datenmodells auf den gleichen Wert. Die Datenfelder des Standarddatenmodells sind `date`, `origdate`, `eventdate` und `urldate`.

`julian=true, false` default: false

Diese Option kontrolliert die Daten vor dem angegebenen Datum, die `gregorianstart`-Option wird automatisch in den Julianischen Kalender umwandeln. Daten, so geändert, gehen zurück zu ‚true‘ für `\ifdatejulian` und `\if<datatype>datejulian` testet (sehen Sie § ??). Bitte beachten Sie, dass

¹⁹<https://www.loc.gov/standards/datetime/pre-submission.html>

Daten, die aus gerade einem Jahr wie ‚1565‘ bestehen, niemals in ein Datum des Julianischen Kalenders umgewandelt werden, da ein Datum ohne einen Monat und Tag eine mehrdeutige Repräsentation im Julianischen Kalender hat²⁰. Zum Beispiel, in dem Fall ‚1565‘, ist das Julianische Jahr ‚1564‘, bis nach ‚10. Januar 1565‘ des Gregorianischen Kalenders, wenn das Julianische Jahr ‚1565‘ wird.

`gregorianstart`= \langle YYYY-MM-DD \rangle

Diese Option kontrolliert das Datum vor den Tagen die in den Julianischen Kalender umgewandelt werden. Es ist ein strenger Formatstring, 4-stelliges Jahr, 2-stelliger Monat und Tag, separariert von einem einzelnen Bindestrichzeichen (jedes gültige Unicodezeichen mit ‚Strich‘-Eigenschaft). Der Standard ist ‚1582-10-15‘, das Datum der Veranlassung des Standard-Gregorianischen Kalenders. Diese Option veranlast nichts, wenn die Option `julian` auf ‚true‘ gesetzt ist.

`datezeros`=true, false default: true

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

`timezones`=true, false default: false

Diese Option steuert, ob die Zeitzonen ausgegeben werden, wenn die Zeiten ausgegeben werden.

`seconds`=true, false default: false

Diese Option steuert, ob Sekunden ausgegeben werden, wenn die Zeiten 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.

`datecirca`=true, false default: false

Diese Option steuert, ob ‚circa‘ zur Ausgabe der Informationen zu den Daten kommt. Wenn sie auf `true` gesetzt ist, werden Daten mit dem `\datecircaprint`-Makro erweitert (§ 3.10.1).

`dateuncertain`=true, false default: false

Diese Option steuert, ob zur Ausgabe Unsicherheitsinformationen zu den Terminen kommen. Wenn sie auf `true` gestzt ist, werden die Daten der Erweiterung durch das `\dateuncertainprint`-Makro folgen (§ 3.10.1).

`dateera`=astronomical, secular, christian default: astronomical

Diese Option steuert, wie Datumsepocheninformationen ausgegeben werden. ‚astronomical‘ verwendet `\dateeraprintpre` zum Ausgeben der Epochen Information vor Start/Enddatums. ‚sekular‘ und ‚christlich‘ nehmen `\dateeraprint`

²⁰Dies ist möglicherweise zutreffend für Daten ohne Zeitangaben, aber dies ist für bibliografische Arbeiten nicht relevant.

zum Ausgeben der Epocheninformation *nach* den Start/Ende/Datums. Standardmäßig führt ‚astronomical‘ zu einem Minuszeichen vor BCE/BC-Daten und ‚secular‘/‚christian‘ führt zu den relevanten Lokalisationsstrings wie ‚BCE‘ oder ‚BC‘ nachr BCE/BC-Daten. Sehen sie dazu die entsprechenden Kommentare in § 3.10.1 und zu den Lokalisationsstrings in § ??.

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

Diese Option setzt das astronomische Jahr, bei denen Epochenlokalisierungsstrings automatisch hinzugefügt werden. Diese Option tut nichts, ohne dass `dateera` auf ‚secular‘ oder ‚christian‘ gesetzt wird.

`time`=12h, 24h, 24hcomp default: 24h

Diese Option steuert das Basisformat der auszugehenden Zeitspezifikationen. Die folgenden Möglichkeiten stehen zur Verfügung:

24h 24-Stundenformat, beispielsweise:

14:03:23
 14:3:23
 14:03:23+05:00
 14:03:23Z
 14:21:23–14:23:45
 14:23:23–14:23:45

24hcomp 24-Stundenformat mit komprimierten Bereichen, beispielsweise:

14:21–23 (Stunden sind die gleichen)
 14:23:23–45 (hStunde und Minute sind die gleichen)

12h 12-Stundenformat mit (lokalisiert) AM/PM-Markern, beispielsweise:

2:34 PM
 2:34 PM–3:50 PM

Wie in den obigen Beispielen ersichtlich ist das tatsächliche Zeitformat sprachspezifisch. Beachten Sie, dass die AM/PM-Zeichenfolge ist eine Reaktion auf die `abbreviate`-Paketoption, wenn dies ein Differenz im spezifischen Ort ergibt. Die führenden Nullen im 24-Stundenformat können separat mit der `timezeros`-Paketoption gesteuert werden. Der Separator zwischen den Zeitkomponenten (`\bibtimesep` und `\bibtzminsep`) und zwischen der Zeit und den beliebigen Zeitzonen (`\bibtimezonesep`) sind auch sprachspezifisch und anpassbar, sehen Sie § 3.10.3. Es gibt globale Paketoptionen, die bestimmen welche Sekunden und Zeitzonen ausgegeben werden (`seconds` und `timezones`, sehen Sie jeweils § 3.1.2.1). Zeitzonen, falls vorhanden, sind entweder ‚Z‘ oder eine positive oder negative Nummer. Kein Standardstil gibt Zeitinformaten aus. Benutzerdefinierte Stile können die zeit ausgeben, wenn `\print, datatype `time`-Befehle genutzt werden, sehen Sie § 4.4.1 (engl. Version).

`labeltime`=12h, 24h, 24hcomp default: 24h

Ähnlich der `time`-Option, steuert aber das Format der Zeiteilfelder, ausgewählt aus dem Feld mit `\DeclareLabeldate`.

`<datatype>time=12h, 24h, 24hcomp` default: 24h

Ähnlich der `time`-Option, steuert aber das Format der Zeiteilfelder, ausgewählt aus dem Feld `<datatype>date` im Datenfeld.

`alltimes=12h, 24h, 24hcomp`

Setzt die `labeltime` und die `<datatype>time`-Option für alle Zeiten im Datenmodell mit den gleichen Werten. Die Datenfelder unterstützenden Zeiteile im Standarddatenmodell sind `date`, `origdate`, `eventdate` und `urldate`.

`dateusetime=true, false` default: false

Gibt an, ob jede Zeitkomponente von einem Datumsfeld nach den Zeitkomponenten auszugeben ist. Der Separator zwischen dem Datum und Zeitkomponenten ist `\bibdatetimesep` aus § 3.10.3. Diese Option tut nichts, wenn ein kompaktes Datumsformat benutzt wird (sehen Sie § 3.1.2.1), da dies sehr verwirrend wäre.

`labeldateusetime=true, false` default: false

Ähnlich der `dateusetime`-Option, steuert aber, ob Zeitkomponenten für das Feld mit `\DeclareLabeldate` zum Ausgeben ausgewählt werden..

`<datatype>dateusetime=true, false`
default: false

Ähnlich der `dateusetime`-Option, steuert aber, ob die Zeitkomponenten für das `<datatype>date`-Feld im Datenmodell ausgegeben werden.

`alldatesusetime=true, false` default: false

Setzt die `labeldateusetime` und die `<datatype>dateusetime`-Option für alle `<datatype>date`-Felder im Datenmodell.

`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 angehängt. Sollte diese Option aktiviert sein, werden die nummerierten Etiketten (also die `labelnumber` wie in § ?? behandelt) zugeordnet, wenn ein Bibliografieeintrag das erste Mal, in jeglicher Bibliografie, ausgegeben wird. Für weitere Erklärungen beachten Sie auch § 3.13.5. Diese Option erfordert zwei LaTeX-Durchläufe, nachdem die Daten exportiert wurden in die `bb1`-Datei mit dem Backend (zusätzlich zu allen anderen Läufen durch Seitenumbrüche etc. erforderlich). Eine wichtige Sache ist zu beachten, wenn Sie den Wert für diese Option in ihrem Dokument ändern (oder den Wert von Optionen, welche von dieser abhängen, solche wie zu dem `\printbibliography`-Makro, sehen Sie § 3.7.2), dann ist es wahrscheinlich, dass Sie die zu ihrer aktuellen Datei gehörige `aux`-Datei löschen müssen und LaTeX erneut durchlaufen lassen müssen, um die korrekte Nummerierung zu erhalten. Sehen Sie auch § 4.1 (engl. Version).

`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 (engl. Version).

`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-eprint-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 von arXiv und elektronische Veröffentlichungsinformationen schauen Sie in § 3.12.7.

`texencoding=auto, <encoding>`

default: auto

Spezifiziert die Verschlüsselung der tex-Datei. Diese Option wirkt sich auf den Datentransfer vom Backend zu BibLaTeX aus. Bei der Verwendung von Biber entspricht diese der Biber `--output_encoding`-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 der bib-Datei. Bei der Verwendung von Biber entspricht diese der `--input_encoding`-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`).

`safeinputenc=true, false`

default: false

Wenn diese Option aktiviert ist, wird BibLaTeX 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 § 2.4.2.1 für die Erläuterung, warum Sie vielleicht dies Option aktivieren.

`bibwarn=true, false` default: true

Im Normalfall wird BibLaTeX automatisch Probleme mit den Daten in der `bib`-Datei, diejenigen die Backends betreffen, wie LaTeX-Probleme übermitteln. Um diese abzustellen, benutzen Sie diese Option.

`mincrossrefs=<integer>` default: 2

Setzt das Minimum von Querverweisen auf `<integer>` wenn ein Backend-Lauf angefordert wird.²¹ Diese Option wirkt sich auch auf den Umgang mit dem Feld `xref` aus. Beachten Sie die Feld-Beschreibung in § 2.2.3 und § 2.4.1 für Details.

`minxrefs=<integer>` default: 2

Wie `mincrossrefs`, aber für `xref`-Felder.

3.1.2.2 Stilspezifik Die folgenden Optionen werden von den Standardstilen (entgegen den Kernpaketen) zur Verfügung gestellt. Technisch gesehen sind sie Präambeloptionen, 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.

3.1.2.3 Intern Die Standardeinstellungen der folgenden Präambeloptionen werden von der Bibliografie 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 (engl. Version) benutzt wird. Dabei stehen folgende Möglichkeiten zur Verfügung.

²¹Sollte ein Eintrag, welcher von anderen Einträgen als Querverweis angegeben wird, in der `bib`-Datei auf einen Grenzwert treffen, wird er in die Bibliografie 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.

<code>true</code>	Aktiviert den automatischen Suchmodus. Das ist dasselbe, wie <code>spread</code> , wenn LaTeX im zweiseitigen Modus ist und bei anderen Fällen wie <code>page</code> .
<code>false</code>	Deaktiviert die Suche.
<code>page</code>	Aktiviert die Suche im Seiten-Modus. In diesem Modus funktioniert die Suche auf einer „pro-Seite“-Basis.
<code>spread</code>	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 (engl. Version).

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

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

<code>true</code>	Aktiviert den Zitierungsschalter auf globaler Ebene.
<code>false</code>	Deaktiviert den Zitierungsschalter.
<code>context</code>	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 (engl. Version). Dabei stehen folgende Möglichkeiten zur Verfügung:

<code>true</code>	Aktiviert den Sucher auf globaler Ebene.
<code>false</code>	Deaktiviert den Sucher.
<code>context</code>	Aktiviert den Sucher im Kontext-sensitivem-Modus. In diesem Modus werden Literaturverweise in Fußnoten und im Text unabhängig von einander gesucht.
<code>strict</code>	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 Eintragschlüssel im Literaturverweismodus angehangen wurde.
<code>constrict</code>	Dieser Modus kombiniert die Eigenschaften von <code>context</code> und <code>strict</code> .

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

`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:

<code>true</code>	Aktiviert Sucher im globalen Modus.
<code>false</code>	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.²²
- `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 (engl. Version).

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

Diese Option kontrolliert den ‚opcit‘-Sucher, was durch den `\ifopcit`-Test aus § 4.6.2 (engl. Version) erforderlich ist. Diese Funktion ist ähnlich dem ‚ibidem‘-Sucher, außer dass die Zitate-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, sehen Sie § 4.11.5 (engl. Version).

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

Diese Option kontrolliert den ‚loccit‘-Sucher, welcher vom `\ifloccit`-Test aus § 4.6.2 (engl. version) benötigt wird. Diese Funktion ist ähnlich der ‚opcit‘-Suche, außerdem wird auch geprüft, ob die `(postnote)`-Argumente übereinstimmen, d. h.

²²Angenommen, das erste Zitat ist „Jones, *Title*; Williams, *Title*“ und das folgende ein „ibidem“. Aus technischer Sicht ist es ziemlich klar, dass ‚ibidem‘ 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.

`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.

`<namepart>inits=true, false` default: false

Wenn diese Funktion aktiviert ist, werden alle `<namepart>`-Namenteile als Initiale erscheinen. Die Option beeinflusst die `\if<namepart>inits-`Tests aus § ???. Die gültigen Namenteile sind im Datenmodell mit dem `\DeclareDatamodelConstant-`Befehl definiert (§ ???).

`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. Die Option arbeitet mit der Redefinierung einiger Makros, die das Format der Initiale kontrollieren. Genaueres in § 3.13.4.

`labelalpha=true, false` default: false

Ob die speziellen Felder `labelalpha` und `extraalpha` zur Verfügung gestellt werden, für Details beachten Sie § 4.2.4 (e. V.) § ???. Diese Option kann diese Option auch auf Per-Typ-Basis gesetzt werden. Siehe auch `maxalphanames` und `minalphanames`. Tabelle 7 summiert die verschiedenen `extra*`-Begriffsklärungszähler und was sie tun.

`maxalphanames=<integer>` default: 3

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

`minalphanames=<integer>` default: 1

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 (e. V.). Diese Option ebenfalls auf Per-Typ-Basis setzbar.

`labeltitle=true, false` default: false

Unabhängig davon ob das spezielle Feld `extratitle` zur Verfügung steht, sehen Sie in § 4.2.4 (e. V.) für weitere Details nach. Beachten Sie, dass das Spezialfeld `labeltitle` immer zur Verfügung steht und diese Option nicht steuert, ob `labeltitle` verwendet wird, um `extratitle`-Informationen generiert werden. Diese Option ist auch auf einer pro-Typ-Basis einstellbar. Tabelle 7 fasst die verschiedenen `extra*`-Begriffserklärungszähler zusammen und was sie beabsichtigen.

Option	Test	Tracks
<code>singletitle</code>	<code>\ifsingletitle</code>	Labelname
<code>uniquetitle</code>	<code>\ifuniquetitle</code>	Labeltitel
<code>uniquebaretitle</code>	<code>\ifuniquebaretitle</code>	Labeltitel, wenn der Labelname Null ist
<code>uniquework</code>	<code>\ifuniquework</code>	Labelname+Labeltitel

Tabelle 6: Arbeit der Vereindeutigungsoptionen

`labeltitleyear=true, false` default: false

Unabhängig davon ob das spezielle Feld `extratitle` zur Verfügung steht, sehen Sie in § 4.2.4 (e. V.) für weitere Details nach. Beachten Sie, dass das Spezialfeld `labeltitle` immer zur Verfügung steht und diese Option nicht steuert, ob `labeltitle` verwendet wird, um `extratitleyear`-Informationen generiert werden. Diese Option ist auch auf einer pro-Typ-Basis einstellbar. Tabelle 7 fasst die verschiedenen `extra*`-Begriffserklärungszähler zusammen und was sie beabsichtigen.

`labeltitleyear=true, false` default: false

Unabhängig davon, ob das spezielle Feld `extratitleyear` zur Verfügung steht, sehen Sie § ?? für Details. Beachten Sie, dass das Spezialfeld `labeltitle` immer zur Verfügung gestellt ist und diese Option steuert diese nicht. `labeltitle` wird verwendet, um eine `extratitleyear`-Information zu erzeugen. Diese Option ist auch auf einer pro-Typ-Basis einstellbar. Tabelle 7 fasst die verschiedenen `extra*`-Eindeutigkeitszähler zusammen und was sie beabsichtigen.

`labeldateparts=true, false` default: false

Unabhängig davon, ob diese Spezialfelder zur Verfügung stehen `labelyear`, `labelmonth`, `labelday`, `labelendyear`, `labelendmonth`, `labelendday`, `labelhour`, `labelendhour`, `labelminute`, `labelendminute`, `labelsecond`, `labelendsecond`, `labelseason`, `labelendseason`, `labeltimezone`, `labelendtimeone` and `extrayear`, sehen Sie § 4.2.4 (e. V.) für Details. Diese Option ist auch einstellbar auf eine pro-Typ-Basis. Tabelle 7 fasst die verschiedenen `extra*`-Eindeutigkeitszähler zusammen und was sie beabsichtigen.

`singletitle=true, false` default: false

Ob oder nicht die erforderlichen Daten bereitstehen für den `\ifsingletitle`-Test, sehen Sie in § 4.6.2 (e. V.) nach weiteren Details. Diese Option ist auch einstellbar auf einer pro-Typ-Basis.

`uniquetitle=true, false` default: false

Ob oder nicht die erforderlichen Daten bereitstehen für den `\ifuniquetitle`-Test, sehen Sie § 4.6.2 (e. V.) für Details. Sehen Sie Tabelle 6 für Details, auf das, was die Daten für diesen Test bestimmt. Diese Option ist auch einstellbar auf einer pro-Typ-Basis.

`uniquebaretitle=true, false` default: false

Ob oder nicht die erforderlichen Daten bereitstehen für den `\ifuniquebaretitle`-Test, sehen Sie § 4.6.2 (e. V.) für Details. Sehen Sie

Tabelle 6 für Details, auf das, was die Daten für diesen Test bestimmt. Diese Option ist auch einstellbar auf einer pro-Typ-Basis.

`uniquework=true, false` `default: false`

Ob oder nicht die erforderlichen Daten bereitstehen für den `\ifuniquework`-Test, sehen Sie § 4.6.2 (e. V.) für Details. Sehen Sie Tabelle 6 für Details auf das, was die Daten für diesen Test bestimmt. Diese Option ist auch einstellbar auf einer pro-Typ-Basis.

`uniqueprimaryauthor=true, false`
`default: false`

Ob oder nicht die erforderlichen Daten bereitstehen für den `\ifuniqueprimaryauthor`-Test, sehen Sie § 4.6.2 (e. V.) für Details.

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

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

<code>true</code>	Ein Alias für <code>full</code> .
<code>false</code>	Abschalten dieses Features.
<code>init</code>	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 `uniquename`-Option auf `unique`list, den `\ifsingletitle`-Test und das `extrayear`-Feld auswirkt. Siehe § 4.11.4 (e. V.) für weitere Details und praktische Beispiele.

`unique`list=true, false, minyear `default: false`

Ob die `unique`list-Zähler aktualisiert werden oder nicht, sehen Sie § 4.6.2 (e. V.) 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 pro-Feld-Basis. Diese Option ist auch auf pro-Typ-Basis setzbar. Möglich sind:

<code>true</code>	Disambiguieren der <code>labelname</code> -Liste.
<code>false</code>	Abschalten dieses Features.

Option	Aktivierte Feld(er)	Aktivierter Counter	Counter-Tracker
labelalpha	labelalpha	extraalpha	label
labeldateparts	labelyear labelmonth labelday labelendyear labelendmonth labelendday labelhour labelminute labelsecond labelendhour labelendminute labelendsecond labelseason labelendseason labeltimezone labelendtimezone	extrayear	labelname+ labelyear
labeltitle	–	extratitle	labelname+labeltitle
labeltitleyear	–	extratitleyear	labeltitle+labelyear

Tabelle 7: Disambiguationscounter

`minyear` Disambiguieren der `labelname`-Liste, nur wenn die verkürzte Liste identisch ist mit einer mit dem gleichen `labelyear`. Dieser Modus eignet sich gut für `author-year`-Stile und erfordert `labelyear=true`.

Beachten Sie, dass die `uniquelist`-Option sich auch auswirkt auf den `\ifsingletitle`-Test und das `extrayear`-Feld. Sehen Sie auch § 4.11.4 (e. V.) für weitere Details und praktische Beispiele.

3.1.3 Eintragsoptionen

Eintragsoptionen sind Paketoptionen, welche bestimmen, wie eine Bibliografie Dateneinträge behandeln wird. Sie können in verschiedenen Bereichen gesetzt werden, die im Folgenden definiert werden.

3.1.3.1 Prämbel/Typ/Eintragsoptionen Die folgenden Optionen sind alle auf pro-Eintrag-Basis oder pro-Eintrag in das `options`-Feld setzbar. Zusätzlich dazu können sie auch als optionales Argument in `\usepackage` so wie in die Konfigurationsdatei und die Dokumentpräambel 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 § 3.5. Diese Option ist auch auf pro-Typ-Basis setzbar.

`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 ein `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 § 3.5. Diese Option ist diese Option auch auf pro-Typ Basis setzbar.

`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: Diese Option ist auch auf pro-Typ-Basis setzbar.

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

Mit den Standardstilen wird `translator` standardmäßig nach dem Titel ausgegeben. Für mehr Informationen beachten Sie auch § 3.5.

`use<name>=true, false`

default: true

Wie mit `useauthor`, `useeditor` und `usetranslator`, alle Namenslisten, mit dem Datenmodell definiert, haben eine Option, die ihr Verhalten bei der Sortierung und Kennzeichnung automatisch steuert. Global pro-Type und pro-Eintrag genannt, `use<name>` wird automatisch erzeugt.

`useprefix=true, false`

default: false

Ob im Standardnamenmodell ein Präfix als Teil eines Namens (von, van, of, da, de, della, usw.) berücksichtigt wird, ist dann,

- wenn der Nachname in den Literaturverweisen ausgegeben wird;
- bei Sortierung;
- bei Erzeugung bestimmter Arten von Etiketten;
- bei Erzeugung von Namen mit einzigartigen Informationen;
- bei Formatierungsaspekten in der Bibliografie.

Beispielsweise, wenn diese Option aktiviert ist, dann geht mit BibLaTeX der Nachname mit dem Präfix voran – Ludwig van Beethoven würde zitiert werden als „van Beethoven“ und alphabetisiert werden als „Van Beethoven, Ludwig“. Ist diese Option jedoch deaktiviert (der Standard), wird er als „Beethoven“ zitiert und als „Van Beethoven, Ludwig“ alphabetisiert. Diese Option kann auch auf pro-Typ-Scopus gesetzt werden. Mit BibLaTeXML-Datenquellen und dem erweiterten

BibTeX-Namensformat (unterstützt von Biber), ist dies auch einstellbar auf pro-Namensliste und pro-Namen-Scopus.

`indexing=true, false, cite, bib`

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

3.1.3.2 Typ/Eintragoptionen Folgende Optionen sind auf einer pro-Type-Basis oder einem pro-Eintrag in dem `options`-Eintrag einstellbar. Sie sind nicht global verfügbar.

`skipbib=true, false` default: false

Wenn diese Option aktiviert ist, wird der Eintrag aus der Bibliografie ausgeschlossen, kann aber immer noch zitiert werden. Diese Option ist auch auf pro-Typ-Basis setzbar.

`skipbiblist=true, false` default: false

Wenn diese Option aktiviert wird, wird der Eintrag davon ausgeschlossen, auch aus der Bibliografieliste. Er ist immer noch in der Bibliografie enthalten und kann auch durch eine Abkürzung etc. zitiert werden. Diese Option ist einstellbar auf einer pro-Type-Basis.

`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 eindeutig sind für den jeweiligen Eintrag und Zitatstil, die die Etiketten benötigen, könnten dabei scheitern, validierte Literaturverweise für die Einträge zu erstellen. Diese Option ist auch auf pro-Typ-Basis setzbar.

`dataonly=true, false` default: false

Diese Option zu setzen entspricht den `uniquename=false`, `uniquelist=false`, `skipbib`, `skipbiblist`, and `skiplab`-Optionen. Normalerweise wird diese Option nicht benötigt. Benutzen Sie sie mit Vorsicht. Diese Option ist auch auf pro-Typ-Basis setzbar.

3.1.3.3 Eintrag-only-Optionen Die folgenden Optionen sind nur per-entry in das `options`-Feld eintragbar. Sie sind nicht global oder per-type verfügbar.

`labelnamefield=<fieldname>`

Besonders ist das Feld zunächst zu prüfen, wenn Sie einen `labelname`-Kandidaten suchen. Es wird im Wesentlichen mit `\DeclareLabelname` nur für diesen Eintrag erstellt.

`labeltitlefield`
`=<fieldname>`

Besonders ist das Feld zunächst zu prüfen, wenn Sie einen `labeltitle`-Kandidaten suchen. Es wird im Wesentlichen mit `\DeclareLabeltitle` nur für diesen Eintrag erstellt.

3.1.4 Vererbungsoptionen

Die folgenden Vererbungsoptionen sind global nur als optionales Argument von `\documentclass` nutzbar oder lokal als optionales Argument von `\usepackage`:

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

3.2 Globale Anwenderanpassung

Unabhängig von der Möglichkeit neue Literaturverweis- und Bibliografistile 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 Bibliografistile 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 aufgelöst:

```
\ExecuteBibliographyOptions[ $\langle$ entry type, ... $\rangle$ ] { $\langle$ key=value, ... $\rangle$ }
```

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

3.3 Standardstile

Dieses Kapitel bietet eine kurze Beschreibung aller Bibliografie- und Literaturverweisstile, die dem BibLaTeX-Paket beiliegen. Wenn Sie ihre eigenen Stile schreiben wollen, schauen Sie in § ?? nach.

3.3.1 Zitierstile

Die Literaturverweisstile, die diesem Paket beiliegen, implementieren einige, verbreitete Zitierstile. Alle Standardstile sorgen für das `shorthand`-Feld und unterstützen Hyperlinks und das Erstellen von Indexen.

`numeric` Dieser Stil beinhaltet ein numerisches Zitierschema, ähnlich dem der bibliografischen Standardeinrichtungen von LaTeX. Er sollte in Verbindung mit den Bibliografistilen, die für die Ausgabe der entsprechenden Etiketten in der Bibliografie 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 Eintragsätze 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 nummeralen 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 Zitierschema 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 Zitierschema 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 Bibliografiestil eingesetzt werden, der die entsprechenden Labels in die Bibliografie 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-Zitierschema aus. Wenn die Bibliografie 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-Bibliografiestil eingesetzt werden, der die entsprechenden Label in der Bibliografie ausgibt. Er ist vorrangig für im-Text-Zitierungen vorgesehen, aber er kann auch für Zitierungen verwendet werden, die in Fußnoten angegeben werden. Der Stil setzt die folgenden Paketoptionen, zur Ladezeit auf:

```
autocite = inline, labeldateparts = true, uniquename = full,
uniquelist=true.
```

authoryear-comp Eine kompakte Variante des `authoryear` Stils, die den Autor nur einmal ausgibt, wenn nachfolgende Referenzen, die an einen einzelnen Zitierbefehl 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, labeldateparts = true,
uniquename=full, uniquelist=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:

```
ibidtracker=constrict. The style will set the following package options
at load time: autocite=inline, labeldateparts=true, uniquename=
full, uniquelist=true, ibidtracker=constrict, pagetracker=
true.
```

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

authoryear-icomp Ein Still der `authoryear-comp` und `authoryear-ibid` kombiniert. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf:

```
kvoptautociteinline, labeldateparts = true, uniquename = full,
uniquelist=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-Zitierungschema 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, uniquename=full, uniquelist=true.
```

authortitle-comp Eine kompakte Variante des `authortitle`-Stils, der den Autor nur einmal ausgibt, wenn nachfolgende Referenzen, die an einen einzelnen Zitierbefehl 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, uniquename = full,
uniquelist=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, uniquename = full, uniquelist = true,
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, uniquename = full, uniquelist = true,
ibidtracker=constrict, pagetracker=true, sortcites=true.
```

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

authortitle-terse Eine prägnante Variante des `authortitle`-Stils, der nur den Titel ausgibt, wenn die Bibliografie 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 = full,
uniquelist=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=full, uniquelist=true.
```

authortitle-ticomp 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 = full,
uniquelist=true. This style also provides an additional preamble option called
ibidpage.
```

Dieser Stil unterstützt eine zusätzliche Präambeloption, mit dem Namen `ibidpage`. Für Details sehen Sie das Stilbeispiel.

verbose Ein „wortreicher“ Zitierungsstil, der eine komplette Zitierung ausgibt, ähnlich mit einem Bibliografieeintrag, 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 bibliografischen 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 sehen Sie das 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 sehen Sie das Stilbeispiel.

verbose-note Dieser Stil ist ähnlich mit dem `verbose`-Stil. Bei diesem wird eine komplette Zitierung ausgegeben ähnlich einem Bibliografieeintrag, 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 Bibliografie 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 bibliografischen 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 sehen Sie das Stilbeispiel.

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 sehen Sie das Stilbeispiel.

verbose-trad1 Dieser Stil führt ein übliches Zitierschema aus. Er ist ähnlich dem `verbose`-Stil. Bei diesem wird eine komplette Zitierung ausgegeben ähnlich einem Bibliografieeintrag, 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 bibliografischen 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 sehen Sie 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 sehen Sie das Stilbeispiel.

verbose-trad3 Noch ein übliches Zitierungsschema. Es ist dem `verbose-trad2` ähnlich, benutzt aber die wissenschaftlichen Abkürzungen, *ebenda* und *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 sehen Sie das Stilbeispiel.

reading Ein Zitierungsstil, der zum Bibliografiestil gehört mit dem gleichen Namen. Er lädt lediglich den `authortitle`-Stil.

Die folgenden Zitierstile 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 Bibliografiestile

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

- numeric** Dieser Stil gibt eine numerische Bezeichnung aus, ähnlich den bibliografischen 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 Paketoptionen 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 sehen Sie das Stilbeispiel.
- 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 Zitierstil vorgesehen. Beachten Sie, dass das `shorthand`-Feld die vorgegebenen Labels aufhebt. Der Stil setzt die folgenden Paketoptionen 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-Zitierstil 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 sehen Sie das Stilbeispiel. Der Stil setzt die folgenden Paketoptionen zur Ladezeit, auf:
- ```
labeldateparts = 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 sehen Sie 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 sehen Sie das Stilbeispiel. Der Stil setzt die folgenden Paketoptionen zur Ladezeit auf: `pagetracker=true`.
- reading** Dieser besondere Bibliografiestil ist für persönliche Leselisten, kommentierte Bibliografien und Anwendungen vorgesehen. Es umfasst wahlweise die Felder `annotation`, `abstract`, `library` und `file` in der Bibliografie. Wenn gewünscht, fügt er verschiedene Arten von kurzen Headers zu der Bibliografie. Dieser Stil unterstützt auch die Präambeloptionen
- ```
entryhead, entrykey, annotation, abstract, library und file,
```
- die kontrollieren, ob die entsprechenden Begriffe in der Bibliografie ausgegeben werden. Für Details sehen Sie das Stilbeispiel. Siehe auch § 3.12.8. 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 Bibliografiestile sind Stile für besondere Zwecke. Sie sind nicht für die Endfassung von einem Dokument vorgesehen:

**draft** Dieser Konzeptstil umfasst die Eintragungsschlüssel in der Bibliografie. Die Bibliografie 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 Verwandte Einträge

Fast alle Bibliografiestile fordern von den Autoren bestimmte Beziehungen zwischen den Einträgen anzugeben: „Reprint of“, „Reprinted in“ etc. Es ist nicht möglich Datenfelder für all diese Relationen zu schaffen. BibLaTeX stellt einen allgemeinen Mechanismus dafür bereit, der die Eintragsfelder `related`, `relatedtype` und `relatedstring` verwendet. Ein verwandter Eintrag muss nicht zitiert werden und erscheint nicht selbst in der Bibliografie (es sei denn, er wird selbst unabhängig zitiert) als ein Clon, der von dem verwandten Eintrag genommen wird wie eine Datenquelle. Das `relatedtype`-Feld sollte eine spezifische Lokalisierungszeichenfolge angeben, die geschrieben werden soll, bevor die die Information aus den verwandten Einträgen geschrieben wird, beispielsweise „Orig. Pub. wie“. Das `relatedstring`-Feld kann verwendet werden, um via `relatedtype` die Zeichenfolge zu überschreiben. Einige Beispiele:

```
@Book{key1,
 ...
 related = {key2},
 relatedtype = {reprintof},
 ...
}

@Book{key2,
 ...
}
```

Hier haben wir die Spezifik, dass der Eintrag `key1` eine Neuauflage des Eintrags `key2` ist. In dem Bibliografietreiber für Book-Einträge, wenn `\usebibmacro{related}`, wird der Eintrag `key1` genannt:

- Wenn der Lokalisierungsstring „reprintof“ definiert wird, wird er im `relatedstring:reprintof`-Format gedruckt. Wenn diese Formatierungsrichtlinie nicht definiert ist, wird die Zeichenfolge im `relatedstring:default`-Format gedruckt.
- Wenn das `related:reprintof`-Makro definiert ist, wird es verwendet, um die Informationen im Eintrag `key2` zu formatieren, ansonsten wird das `related:default`-Makro verwendet.

- Wenn das `related:reprintof`-Format definiert ist, wird es verwendet, um die lokalisierungszeichenfolge und die Daten zu formatieren. Wenn dieses Format nicht definiert ist, dann wird stattdessen das `related`-Format verwendet.

Es wird auch eine Kaskadierung und/oder kreisförmige Beziehung unterstützt:

```
@Book{key1,
 ...
 related = {key2},
 relatedtype = {reprintof},
 ...
}

@Book{key2,
 ...
 related = {key3},
 relatedtype = {translationof},
 ...
}

@Book{key3,
 ...
 related = {key2},
 relatedtype = {translatedas},
 ...
}
```

Mehrere Beziehungen zum selben Eintrag sind möglich:

```
@MVBook{key1,
 ...
 related = {key2,key3},
 relatedtype = {multivolume},
 ...
}

@Book{key2,
 ...
}

@Book{key3,
 ...
}
```

Beachten Sie die Reihenfolge der Schlüssel in den Listen aufeinander bezogener Einträge; dies ist wichtig. Die Daten von mehreren aufeinander bezogenen Einträgen werden in der Reihenfolge der aufgeführten Schlüssel in diesem Feld gedruckt. Sehen Sie § ?? für weitere Details zu den Mechanismen hinter diesen Merkmalen. Sie können diese Funktionen deaktivieren, indem Sie die Paketoption `related` aus § 3.1.2.1 nehmen.

Sie können die `relatedoptions`-Option auf die Optionen der verwandten Eintragsdatenklone setzen. Die ist nützlich, wenn Sie die `dataonly`-Option überschreiben müssen, die standardmäßig auf alle verwandten Eintragsklone eingerichtet ist. Beispielsweise, wenn Sie von den Namen einige in den verwandten Klonen in ihrem Dokument aussetzen wollen, Sie können die `namen` in den anderen Einträgen vereindeutigt haben, aber normalerweise wird dies nicht über verwandte Klone geschehen, die `pro-Eintrag-dataonly`-option wird gesetzt und dies wiederum setzt `uniquename = false` und `uniquelist = false`. In solch einem Fall können Sie `relatedoptions` setzen, nur `skiplab`, `skipbib`, `skiplos/skipbiblist`.

### 3.5 Sortierungsoptionen

Dieses Paket unterstützt verschiedene Sortierungsschemata für Bibliografien. Das Sortierungsschema wird mit der Paketoption `sorting` aus § 3.1.2.1 ausgewählt. Abgesehen von den normalen Datenfeldern gibt es auch besondere Felder, die benutzt werden können, um das Sortieren der Bibliografie zu verbessern. C.1 und C.2 (e. V.) geben einen Entwurf des alphabetischen Schema an, das von BibLaTeX unterstützt wird. Chronologische Sortierungsschemata sind in C.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 Namensfelder 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 C.3 dargestellt sind, einen zusätzlichen Begriff enthalten: `labelalpha` ist das Label, benutzt vom ‚alphabetischen‘ Bibliografiestil. Genau genommen ist die Reihung, die für das Sortieren benutzt wird, `labelalpha + extraalpha`. Die Sortierungsschemata in appendix ?? sind vorgesehen, nur in Zusammenhang mit alphabetischen Stilen benutzt zu werden.

Die chronologischen Sortierungsschemata, aufgeführt in appendix ??, 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 `yndt`-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 Bibliografie 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.

### 3.6 Daten-Annotationen

Idealerweise sollten in einer Bibliografiedatendatei keine Formatierungsinformationen sein, jedoch scheint es manchmal so, dass es eine fragwürdige Praxis ist, dies als einzigen Weg für die gewünschten Ergebnisse anzusehen. Daten-Annotationen sind ein Weg dies anzugehen, indem Nutzer semantische Informationen anfügen (und nicht als typografische Markups), Informationen in einer Bibliografiedatenquelle, so dass die Information genutzt werden kann für das Markup durch einen Stil. Beispielsweise, wenn man in einem Werk bei einem bestimmten Namen markieren möchte, ob es sich um einen Studentenautor (angezeigt durch ein hochgestelltes Sternchen) oder einen korrespondierenden Autor (angezeigt durch Fettdruck) handelt, dann könnte man es folgendermaßen versuchen:

```
@MISC{Article1,
 AUTHOR = {Last1*, First1 and \textbf
 ↪ {Last2}, \textbf{First2} and Last3, First3}
}
```

Damit gibt es mehrere Probleme. Zum einen werden damit BibTeX's fragile Namensparsing-Routinen durchbrochen und wahrscheinlich dieses gar nicht kompilieren. Zum anderen ist es nicht nur das Vermischen von Daten mit Markup, es wird so ein hartcodierter Vorgang: Diese Daten können nicht leicht geteilt und mit anderen Stilen verwendet werden. Dagegen ist es möglich, diese Formatierung mit BibLaTeX-Einbauten in einen Stil oder in einem Dokument zu erreichen, dies ist eine komplexes und unzuverlässiges Verfahren, das wir nicht vielen Benutzer für die Benutzung wünschen.

Um diese Probleme zu lösen, hat BibLaTeX eine allgemeine Datenannotationseinrichtung, die ermöglicht, mit einer Komma-separierten Liste von Datenanmerkungen in Datenfeldern aufzulisten, Einträge in Datenfeldlisten (wie Namen) und sogar Teile von spezifischen Einträgen, wie Teile von Namen (Vorname, Familienname etc.). Es gibt Makros, entwickelt zur Überprüfung von Annotationen, die in Formatierungsanweisungen verwendet werden können.

Es gibt drei „scopes“ für Daten-Annotationen, angeordnet nach zunehmender Spezifik:

- `field`-angelegt für „top-level“-Felder in einem Datenquell-Eintrag.
- `item`-angelegt für Einträge eines Listenfelds in einem Datenquell-Eintrag.

- part-angelegt für Teile von Einträgen eines Listenfelds in einem Datenquell-Eintrag.

Daten-Annotationen werden supportet für BibTeX- und BibLaTeXXML-Datenquellen.

Annotationen an Daten anzuhängen ist in BibLaTeXXML-Datenquellen einfach, wenn sie durch einfache XML-Attribute festgelegt sind. Bezugnehmend auf das obige Beispiel, haben wir:

```
<bltx:entries xmlns:bltx="http://biblatex-biber.
 ↳ sourceforge.net/biblatexml">
 <bltx:entry id="test" entrytype="misc">
 <bltx:names type="author">
 <bltx:name>
 <bltx:namepart type="given" initial="F">First1</
 ↳ bltx:namepart>
 <bltx:namepart type="family" initial="L"
 ↳ annotation="student">Last1</bltx:namepart>
 </bltx:name>
 <bltx:name annotation="corresponding">
 <bltx:namepart type="given" initial="F">First2</
 ↳ bltx:namepart>
 <bltx:namepart type="family" initial="L">Last2</
 ↳ bltx:namepart>
 </bltx:name>
 <bltx:name>
 <bltx:namepart type="given" initial="F">First3</
 ↳ bltx:namepart>
 <bltx:namepart type="family" initial="L">Last3</
 ↳ bltx:namepart>
 </bltx:name>
 </bltx:names>
 </bltx:entry>
</bltx:entries>
```

Hier werden die Annotationen in naheliegender Weise an die Dateneinträge angehängt. Mit BibTeX-Datenquellen ist das Format für Annotationen nicht ganz so intuitiv.

```
@MISC{ann1,
 AUTHOR = {Last1, First1 and Last2, First2 and Last3
 ↳ , First3},
 AUTHOR+an = {1:family=student;2=corresponding},
}
```

Hier ist das Feldnamen-Suffix +an ein Nutzer-definiertes<sup>23</sup> Suffix, das ein Datenfeld als Annotation eines nichtsuffixierten Felds markiert. Das Format von Annotationsfeldern in BibTeX-Datenquellen ist wie folgt:

```
<annotationspecs> ::= <annotationspec> [";" <
 ↳ annotationspec>]
```

<sup>23</sup>Beachten Sie Biber s-Annotations-marker-option.

```

<annotationspec> ::= [<itemcount> [":" <part>]] "="
 ↪ <annotations>
<annotations> ::= <annotation> ["," <annotation>]
<annotation> ::= (string)

```

Das heißt, eine oder mehrere Spezifikationen getrennt durch Semikolons. Jede Spezifikation hat ein Gleichheitszeichen und es folgt eine Komma-getrennte Liste von Annotationsschlüsselwörtern. Um ein spezifisches Item in einer Liste zu annotieren, legen Sie die Zahlen der Listeneinträge vor dem Gleichheitszeichen fest (die Liste beginnt mit 1). Wenn Sie einen spezifischen Teil des Listenitems annotieren wollen, geben Sie seinen Namen nach der Listenitemnummer an, vorausgehend ein Doppelpunkt. Namen von Teilnamen werden im Datenmodell definiert, sehen Sie § 4.2.3 (e. V.) Einige Beispiele:

```

AUTHOR = {Last1, First1 and Last2, First2 and Last3
 ↪ , First3},
AUTHOR+an = {3:given=annotation1, annotation2},
TITLE = {A title},
TITLE+an = {=a title annotation, another title
 ↪ annotation},
LANGUAGE = {english and french},
LANGUAGE+an = {1=annotation3; 2=annotation4}
}

```

Um auf die Annotationsinformation bei der Formatierung der Bibliografiedaten zuzugreifen, werden drei Makros zur Verfügung gestellt, korrespondierend mit den drei Annotationsbereichen:

`\iffieldannotation{<annotation>}{<true>}{<false>}`

Wählt *<true>*, wenn das aktuelle Datenfeld eine Annotation hat *<annotation>* und *false* andernfalls.

`\ifitemannotation{<annotation>}{<true>}{<false>}`

Wählt *<true>*, wenn das aktuelle Item in dem aktuellen Datenfeld eine Annotation hat *<annotation>* und *false* andernfalls.

`\ifpartannotation{<part>}{<annotation>}{<true>}{<false>}`

Wählt *<true>*, wenn der Teilnamen *<part>* des aktuellen Items in dem aktuellen Datenfeld eine Annotation hat *<annotation>* und *false* andernfalls.

Diese Makros sind an den gleichen Stellen wie `\currentfield`, `\currentlist` und `\currentname` (sehen Sie § 4.4.2 (e. V.)), das heißt innerhalb der Formatierungsanweisungen. Sie bestimmen automatisch den Namen des aktuellen Datenfelds, das verarbeitet wird, und auch den aktuellen `listcount`-Wert, der den aktuellen Eintrag in Listenfeldern bestimmt. Teile, sowas wie Namensteile, müssen explizit benannt werden. Ein Beispiel dafür, wie die Annotationsinformation zu verwenden ist, wird, um das eingangs im Kapitel präsentierte Problem zu lösen: Dies könnte verwendet werden bei den Richtlinien der Namensformatierung, für das Setzen von Sternchen nach allen Namen, die als „Student“ annotiert werden sollen:

```
\ifpartannotation{family}{student}
{*}
}{%}
```

Um Familiennamen aus Namenslisteneinträgen mit Fettdruck als „korrespondierend“ zu markieren:

```
\renewcommand*{\mkbibnamegiven}[1]{%
\ifitemannotation{corresponding}
{\textbf{#1}}
{#1}}

\renewcommand*{\mkbibnamefamily}[1]{%
\ifitemannotation{corresponding}
{\textbf{#1}}
{#1}}
```

## 3.7 Bibliografiebefehle

### 3.7.1 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 einer Ressource ein Label zu. Der *identifier* kann an Stelle des vollständigen Ressourcenamens im optionalen Argument von `refsection` verwendet werden (siehe § 3.7.4).

`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

Der 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.

`biblatexml` Experimental XML-Format für BibLaTeX. Sehen Sie D.

`\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` (§ 3.7.4) 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>` zu keiner Ressourceliste hinzugefügt werden. Dies wird nur für Ressourcen benötigt, die 1) ausschließlich in den optionalen Argumente von `refsection` (3.7.4) stehen und 2) Optionen benötigen, die von den vorgegebenen Einstellungen abweichen. In diesen Fall kommt `\addsectionbib` zum Einsatz, um die `<resource>` vorher zu markieren für das 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 bibliografischen Ressourcen, er 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, mehrere Dateien hinzuzufügen, ist erlaubt. Dieser Befehl wird missbilligt. Bitte überlegen Sie stattdessen, `\addbibresource` zu benutzen.

### 3.7.2 Die Bibliografie

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

Dieser Befehl gibt die Bibliografie aus. Er 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 Bibliografie 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`. Sehen Sie auch §§ 3.7.3 and 3.7.7.

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

Die Bibliografie 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`. Sehen Sie auch §§ 3.7.3 and 3.7.7.

`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 sehen Sie § 3.7.7.

`prenote=<name>`

Die Prenote (Vornotiz) ist ein beliebiges Stück Text, das nach der Überschrift ausgegeben wird, aber von vor der Liste von Referenzen stammt. Diese Option wählt die Prenote `<name>` aus, wie in `\defbibnote` definiert. Automatisch wird keine Prenote 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 sehen Sie § 3.7.8.

`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 sehen Sie § 3.7.8.

`section=<integer>` default: `current section`

Gibt nur Einträge aus, die in der Referenzsektion `<integer>` 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 sehen Sie § 3.7.4 und für Anwendungsbeispiele § 3.12.3.

`segment=<integer>` default: 0

Gibt nur Einträge aus, die im Referenzsegment `<integer>` 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 sehen Sie § 3.7.5 und für Anwendungsbeispiele § 3.12.3.

Denken Sie daran, dass die Segmente in einem Abschnitt inbezug auf den Abschnitt lokal nummeriert werden, so dass das Segment, das Sie anfordern, das n-te Segment im gewünschten (oder gerade aktive einschließend) Abschnitt sein wird.

`type=<entrytype>`

Gibt nur Einträge aus, deren Eintragtyp `<entrytype>` ist.

`nottype=<entrytype>`

Gibt nur Einträge aus, deren Eintragstyp nicht `<entrytype>` ist. Diese Option kann mehrere Male benutzt werden.

`subtype=<subtype>`

Gibt nur Einträge aus, deren `entrysubtype` definiert ist und `<subtype>` ist.

`notsubtype=<subtype>`

Gibt nur Einträge aus, deren `entrysubtype` undefiniert ist und nicht `<subtype>` ist. Diese Option kann mehrere Male benutzt werden.

`keyword=<keyword>`

Gibt nur Einträge aus, deren `keywords` Feld ein `<keyword>` mit einschließt. Diese Option kann mehrere Male benutzt werden.

`notkeyword=<keyword>`

Gibt nur Einträge aus, deren `keywords` kein `<keyword>` 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>`

Filtert die Einträge mit dem Filter `<name>`, wie er mit `\defbibfilter` definiert ist. Für Details sehen Sie § 3.7.9.

`check=<name>`

Filtert die Einträge mit dem Check `<name>`, wie er mit `\defbibcheck` definiert ist. Für Details sehen Sie § 3.7.9.

`resetnumbers=true, false`

Diese Option ist nur bei numerischen Zitierungen/Bibliografistilen 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 Bibliografie zugeteilt werden, z. B. wird die Nummerierung 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/Bibliografistilen 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 Bibliografie zuweisen. Dies ist nützlich, wenn eine numerischen Bibliografie mit einer oder mehreren Subbibliografien, die ein anderes 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 Sektion 0 zugeordnet sind. Für Anwendungsbeispiele sehen Sie § 3.12.3. 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 sehen Sie § 3.12.3. 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 Bibliografiekategorien 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 sehen Sie § 3.12.3. Die Optionen sind eine Teilmenge von denen, die `\printbibliography` unterstützt. Gültige Optionen sind `env`, `prenote`, `postnote`, `maxnames`, `minnames`, `maxitems`, `minitems` und `section`. Beachten Sie, 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 (sehen Sie § 3.7.10).

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

Dieser Befehl gibt einen Bibliografietitel aus, der mit `\defbibheading` definiert wurde. Er erfordert eine optionales Argument, das eine Liste von Optionen ist, die in `<key>=<value>`-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 sehen Sie § 3.7.7, Für Anwendungsbeispiele auch §§ 3.12.3 and 3.12.4.

Um eine Bibliografie mit einem anderen Sortierschema als dem globalen auszugeben, wählen Sie die Bibliografie-Wechsel-Befehle aus § 3.7.10.

### 3.7.3 Bibliografielisten

BibLaTeX kann zusätzlich zum Ausgeben von normalen Bibliografien auch beliebige Listen von Informationen ausdrucken. Diese sind abgeleitet aus den Bibliografie-daten, wie von der Abkürzungskürzel für bestimmte Einträge oder einer Liste der Abkürzungen von Zeitschriftentiteln.

Eine Bibliografieliste unterscheidet sich von einer normalen Bibliografie darin, dass die gleichen Bibliografietreiber alle Einträge nehmen für die Ausgabe, anstatt einen speziellen Treiber für jede Eintragsableitung, je nach Art des Eintrags, zu verwenden.

```
\printbiblist[⟨key=value, ...⟩]{⟨biblistname⟩}
```

Dieser Befehl gibt eine Bibliografieliste aus. Er nimmt ein optionales Argument, das aus einer Optionenliste in  $\langle key \rangle = \langle value \rangle$ -Notation stammt. Gültige Optionen sind alle Optionen, die von `\printbibliography` unterstützt werden (§ 3.7.2), außer `prefixnumbers`, `resetnumbers` und `omitnumbers`. Wenn es irgendwelche `refsection`-Umgebungen im Dokument sind, wird die Bibliografieliste für diese Umgebungen lokal sein; sehen Sie § 3.7.4 für Details. Standardmäßig verwendet dieser Befehl die Überschrift `biblist`. Sehen Sie § 3.7.7 für Details.

Der  $\langle biblistname \rangle$  ist ein obligatorisches Argument, welches die Bibliografie benennt. Dieser Name wird zum Identifizieren verwendet:

- Der Standardbibliografietreiber wird verwendet, um die Eintragsliste auszugeben.
- Ein Standardfilter wird mit `\DeclareBiblistFilter` deklariert (sehen Sie § 4.5.7 (e. V.), um die ausgefilterten Einträge an Bibler zu geben.
- Eine Standardprüfung wird mit `\defbibcheck` deklariert (sehen Sie § 3.7.9), um die Listeneinträge nachzubearbeiten.
- Die Standard-bib-Umgebung zu nehmen.
- Das Standardsortierschema für Namen zu nehmen.

In Bezug auf die Sortierung der Liste: Der Standard der Sortierung ist, das benannte Sortierungsschema nach der Bibliografieliste zu verwenden (falls es existiert) und erst dann, wenn es zurückfällt auf das aktuelle Kontextsortierschema, dies ist nicht definiert (sehen Sie § 3.7.10).

Die häufigste Bibliografieliste ist eine Küzelabkürzungsliste für bestimmte Einträge und so hat diese einen Komfort mit `\printshorthands[...]` für die Abwärtskompatibilität, die wie folgt definiert ist:

```
\printbiblist[...]{shorthand}
```

BibLaTeX bietet eine automatische Unterstützung für Datenquellenfelder, die im Standarddatenmodell markiert sind mit ‚Label fields‘ (Sehen Sie § 2.2.2). Solche Felder sind automatisch definiert für Sie:

- Eine Standard-bib-Umgebung (sehen Sie § 3.7.7).
- Ein Bibliografielistenfilter (sehen Sie § 4.5.7 (e. V.)).
- Einige unterstützte Formate und Längen (sehen Sie § 4.10.5 und § 4.10.4).

Daher ist nur ein minimales Setup nötig, um Listen mit solchen Feldern auszugeben. Zum Beispiel, um eine Liste von Abkürzungen für Zeitschriften auszugeben, können Sie die einfach in ihre Präambel setzen:

```
\DeclareBibliographyDriver{shortjournal}{%
 \printfield{journaltitle}}
```

Sie können Sie auch in ihr Dokument setzen, wenn Sie diese Liste ausgeben möchten:

```
\printbiblist[title={Journal Shorthands}]{shortjournal}
```

Weil `shortjournal` in dem Standarddatenmodell als ein ‚Label field‘ definiert ist, gibt es diese Beispiele:

- Das Nehmen der automatisch erzeugten ‚shortjournal‘-bib-Umgebung.
- Das Nehmen der automatisch erzeugten ‚shortjournal‘-Bibliografie-Filterliste, um nur Einträge mit einem `shortjournal`-Feld in die `.bbl` zurückzuführen.
- Das Nehmen der definierten ‚shortjournal‘-Bibliografietreiber, um Einträge auszugeben.
- Das Nehmen der Standard-‚biblist‘-Überschriftenheading, jedoch überschreibend die Titel mit ‚Journal Shorthands‘
- Das Nehmen des aktuellen Bibliografie-Kontextsortierschemas, wenn kein Schema mit dem Namen `shortjournal` existiert.

Öfters werden Sie automatisch abgeholt, wenn Sie ein Labelfeld der Liste sortieren wollen und es nach der Liste benannt wird, in diesem Fall können Sie einfach Folgendes tun:

```
\DeclareSortingScheme{shortjournal}{
 \sort{
 \field{shortjournal}
 }
}
```

Natürlich können alle Standards mit Optionen zu `\printbiblist` überschrieben werden und auch Definitionen von Umgebungen, Filtern etc. Auf diese Weise können beliebige Typen von Bibliografielisten ausgegeben werden, die eine Vielzahl von Informationen der Bibliografiedaten enthalten.

Bibliografielisten werden oft zum Ausgeben von verschiedenen Arten von Kürzeln verwendet und dies kann zu doppelten Einträgen führen, wenn mehr als ein Eintrag das gleiche Kürzel hat. Beispielsweise würden einige Zeitschriftenartikel in der gleichen Zeitschrift in der Liste der Zeitschriftenkürzel zu doppelten Einträgen führen. Sie können die Tatsache nutzen, dass solche Listen ein automatisches Einlesen eines `\bibcheck` der gleichen Namen vornimmt, wie die Liste definiert einen Check definiert, um Duplikate zu entfernen. Wenn Sie eine Liste definieren um die Zeitschriftenkürzel auszugeben, nehmen Sie das `shortjournal`-Feld, Sie könnten einen `\bibcheck` folgendermaßen definieren:

```

\defbibcheck{shortjournal}{%
 \iffieldundef{shortjournal}
 {\skipentry}
 {\iffieldundef{journal}
 {\skipentry}
 {\ifcsdef{\strfield{shortjournal}=\strfield{
↪ journal}}
 {\skipentry}
 {\savefieldcs{journal}{\strfield{shortjournal
↪ }=\strfield{journal}}}}}}

```

### 3.7.4 Bibliografiesektionen

Die `refsection`-Nachbarschaft wird im Dokument benutzt, um eine Referenzsektion zu markieren. Diese Nachbarschaft ist nützlich, wenn man separate, unabhängige Bibliografien 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. Sehen Sie die `refsection`-Paketooptionen in § 3.1.2.1 als einen Weg an, diese zu automatisieren. Für Anwendungsbeispiele sehen Sie auch § 3.12.3.

```

\begin{refsection}[\langle resource, ... \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 Ressourcenlisten benutzen, wie sie mit `\addbibresource` in der Präambel festgelegt werden. Wenn das Argument bereitgestellt ist, ersetzt es die vorgegebene Ressourcenliste. 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 Bibliografie ausgegeben wird und die dazugehörige Option von `\printshorthands`, wenn die Liste von Kürzeln ausgegeben wird. Bibliografiesektionen sind nummeriert, beginnend bei 1. Die Nummer der aktuellen Sektion wird auch in die Kopiedatei 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 `\printbiblist`.

```

\newrefsection[\langle resource, ... \rangle]

```

Dieser Befehl ist der `refsection`-Nachbarschaft ähnlich, außer dass 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.

### 3.7.5 Bibliografiesegmente

Die `refsegment`-Nachbarschaft wird im Dokument benutzt, um ein Referenzsegment zu markieren. Diese Nachbarschaft ist nützlich, wenn man eine allgemeine Bibliografie will, die in Kapitel, Sektionen oder jeden anderen Teil des Dokuments unterteilt wird. Eigentlich sind Referenzsegmente komplett unabhängig von den Dokumentteilen, wie etwa `\chapter` und `\section`, obwohl sie speziell für Kapitel oder Sektion benutzt werden. Sehen Sie die `refsegment`-Paketoptionen in § 3.1.2.1 als einen Weg an, diese zu automatisieren. Für Anwendungsbeispiele sehen Sie auch § 3.12.3.

```
\begin{refsegment}
\end{refsegment}
```

Der Unterschied zwischen einer `refsection`- und einer `refsegment`-Nachbarschaft ist, dass die erstere Labels erzeugt, die lokal zu der Nachbarschaft sind, während die 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. Benutzen Sie die `refsegment`-Option von `\printbibliography`, um ein Segment auszuwählen, wenn die Bibliografie 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 er eher ein selbstständiger Befehl ist als eine Nachbarschaft. Er beendet automatisch das bisherige Referenzsegment (gegebenenfalls) und beginnt sofort ein neues. Beachten Sie, dass das Referenzsegment, das beim letzten `\newrefsegment`-Befehl startet, bis zum Ende des Dokuments erweitert wird. Benutzen Sie `\endrefsegment`, wenn man es früher beenden will.

### 3.7.6 Bibliografiekategorien

Bibliografiekategorien ermöglichen es, die Bibliografie in mehrere Teile, die verschiedenen Themen oder verschiedenen Typen von Referenzen gewidmet sind, zu teilen, zum Beispiel in primäre und sekundäre Quellen. Für Anwendungsbeispiele sehen Sie § 3.12.4.

```
\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 das Dokumentgerüst geschrieben werden. Der `<key>` kann entweder ein einzelner Eingabeschlüssel sein oder eine durch Kommata getrennte Liste von Schlüsseln. Die Verknüpfung ist global.

### 3.7.7 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 Bibliografie 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 (e. V.)). 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 Teilabschnittkopf 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 (e. V.)). Für klassenspezifische Anleitungen siehe ebenfalls §§ 3.13.1 und 3.13.2.

### `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 `\printbiblist`, sind vordefiniert:

`biblist`

Das ist die Standardüberschrift von `\printbiblist`, wenn die `heading`-Option nicht gegeben wird. Sie ist ähnlich mit `bibliography`, außer dass die Lokalisierungstringa `shorthands` anstelle von `bibliography` oder `references` verwendet wird (sehen Sie § 4.9.2 (e. V.), auch §§ 3.13.1 und 3.13.2 für klassenspezifische Hinweise).

`biblistintoc`

Ähnlich mit `biblist` von oben, aber ein Eintrag ins Inhaltsverzeichnis wird erstellt.

`biblistnumbered`

Ähnlich mit `biblist` von oben, aber genommen, um für `\chapter` oder `\section` eine nummerierte Überschrift, die auch ins Inhaltsverzeichnis gefügt wird, zu schaffen.

### 3.7.8 Bibliografieanmerkungen

```
\defbibnote{⟨name⟩}{⟨text⟩}
```

Definiert die Bibliografie-notiz `⟨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. Sehen Sie ebenfalls § 3.13.6.

### 3.7.9 Bibliografiefilter 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 Referenzsegment `⟨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 `\addtocategory` 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. Sehen Sie in die 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 Bibliografiefilter <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 Bibliografiedaten 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}
 {\ifnumless{\thefield{year}}{2000}
 {\skipentry}
 {}}
 {\skipentry}}

```

Siehe in das Autorenhandbuch, insbesondere §§ 4.6.2 und 4.6.3 (e. V.) weitere Details.

### 3.7.10 Referenzkontexte

Referenzen in einer Bibliografie sind zitiert und ausgegeben in einem ‚Kontext‘. Der Kontext bestimmt die Daten, die tatsächlich verwendet werden beim Zitieren oder bibliografische Daten, die für einen Eintrag zur Verfügung stehen. Ein Kontext besteht aus den folgenden Informationen (Das ‚Kontext‘-Konzept ist als zukünftig erweiterbar entworfen):

- Ein Sortierschema,
- ein Schema für das Erstellen von Sortierschlüsseln für Namen,
- ein string-Präfix-Schema, das alphabetische oder numerische Labels verwendet.

Der Bibliografiekontext-Punkt ist doppelt. Zum einen werden Kontexte verwendet, um Optionen zu setzen, die eine ausgegebene Bibliografie beeinflussen. Zum anderen, um die Datenausgabe durch Zitierbefehle zu beeinflussen. Die frühe Nutzung ist die häufigste, wenn man mehr als eine Bibliografieliste mit Unterschieden ausgeben muss, um Beispiel beim Sortieren.

```

\usepackage[sorting=nyt]{biblatex}
\begin{document}
\cite{one}
\cite{two}
\printbibliography
\newrefcontext[sorting=ydnt]
\printbibliography

```

Hier geben wir zwei Bibliografien aus, eine mit dem Standardsortierschema ‚nyt‘ und eine mit dem ‚ydnt‘-Sortierschema.

Um die zweite Art der Nutzung des Bibliografiekontextes zu demonstrieren, müssen wir verstehen, dass die tatsächlichen Daten für einen Eintrag je nach Kontext variieren. Am deutlichsten wird dies im Falle der *extra\**-Felder wie *extrayear*, die vom Backend entsprechend der Reihenfolge der Einträge *nach* der Sortierung generiert werden, so dass sie in der erwarteten Folge ‚a, b, c‘ erscheinen. Dies zeigt deutlich, dass die *Daten* eines Eintrags unterschiedlich zum Sortierschema sein können. Wenn ein Dokument mehr als eine Bibliografieliste mit verschiedenen Sortiersystemen enthält, kann es passieren, dass die *.bbl* Listen mit der gleichen Eintragsortierung aber unterschiedlichen Daten enthält (einen anderen Wert für *extrayear*, beispielsweise). Der Zweck der Bibliografiekontexte ist, die Dinge in einem Kontext zu kapseln, so dass BibLaTeX die korrekten Eintragsdaten verwenden

kann. Ein Beispiel ist die Ausgabe einer Bibliografieliste mit einer anderen Sortierreihenfolge für die globale Sortierreihenfolge, wo die `extra*`-Felder unterschiedlich sind zwischen Sortierlisten mit dem gleichen Eintrag:

```
\usepackage[sorting=nyt,style=authoryear]{biblatex}
\DeclareSortingScheme{yntd}{
 \sort{
 \field[strside=left,strwidth=4]{sortyear}
 \field[strside=left,strwidth=4]{year}
 \literal{9999}
 }
 \sort{
 \field{sortname}
 \field{author}
 \field{editor}
 }
 \sort[direction=descending]{
 \field{sorttitle}
 \field{title}
 }
}
\begin{document}
\cite{one}
\cite{two}
\printbibliography
\newrefcontext[sorting=yntd]
\cite{one}
\cite{two}
\printbibliography
```

Hier nun, die zweite Verwendung der Zitate, zusammen mit dem `\printbibliography`-Befehl werden Daten aus dem Kontext des benutzerdefinierten Sortierschema `'yntd'` verwendet, das auch unterschiedlich zu den Daten im Zusammenhang mit dem Standardschema `'nyt'`. Das heißt, die Zitierlabels (in einem `authoryear`-Stil, der `extrayear` nimmt) können Unterschiede *für die genau gleichen Einträge* zwischen verschiedenen Bibliografiekontexten sein und so können die Zitate selbst anders aussehen.

Standardmäßig werden Daten für eine Zitierung vom Referenzkontext der letzten Bibliografie, in welcher sie ausgegeben wurden, geholt. Beispielsweise in das Fragment:

```
\begin{document}

\cite{book, article, misc}

\printbibliography[type=book]

\newrefcontext[labelprefix=A]
\printbibliography[type=article]

\newrefcontext[sorting=ydnt, resetnumbers]
```

```
\printbibliography[type=misc]
\end{document}
```

Dieses Beispiel zeigt auch die Deklaration und das Benutzen von benannten Referenzkontexten. Unter der Annahme, dass ihre Eingabetasten eingerichtet sind auf ihre Eintragsstypen, ist dies die Standardsituation fürs Zitieren, die normalerweise den Erwartungen der Nutzer entsprechen:

- Das Zitat des Eintrags `book` würde seine Daten aus dem globalen Referenzkontext entnehmen, weil die letzte Bibliografie, in der es ausgegeben wurde, der globale Referenzkontext war.
- Das Zitat des Eintrags `article` würde seine Daten aus dem Referenzkontext mit dem `labelprefix=A` entnehmen und würde daher ein ‚A‘ Präfix haben, wenn es zitiert wird.
- Das Zitat des Eintrags `misc` würde seine Daten aus dem Referenzkontext mit `sorting=ydnt`.

In Fällen, in denen die Benutzer Einträge haben, die in mehreren Bibliografien in verschiedenen Formen oder mit potentiell unterschiedlichen Labels vorkommen (in einem numerischen Schema mit verschiedenen `labelprefix` Werten beispielsweise), kann es notwendig sein, BibLaTeX zu sagen, aus welchem Kontext man die Zitate entnehmen möchte. Wie oben gezeigt, kann dies durch Zitate, die explizit innerhalb der Referenzkontexte gesetzt wurden. Dies kann in einem großen Dokument belastend sein und so gibt es eine spezifische Funktionalität für die programmatische Zuordnung der Zitate zu Referenzkontexten, sehen Sie zu den `\assignrefcontext*`-Makros unten.

```
\begin{refcontext}[<key=value, ...>]
\end{refcontext}
```

Packt ein Bibliografiekontextumgebung ein. Die Optionen definieren die Kontextattribute. Alle Kontextattribute sind optional und standardmäßig bei den globalen Einstellungen nicht vorhanden. Die aktuelle Optionen sind:

```
sorting=<name>
```

Spezifizierung eines Sortierschemas, das zuvor mit `\DeclareSortingScheme` definiert wurde. Dieses Schema wird genommen, um Daten für einen Eintrag in den Befehlen innerhalb des Kontexts zu bestimmen, abzurufen und/oder auszugeben.

```
sortingnamekeyscheme=<name>
```

Spezifizierung einer Namen-Schlüsselschema-Sortierung, die zuvor mit `\DeclareSortingNamekeyScheme` definiert wurde. Dieses Schema wird verwendet, Sortierungsschlüssel innerhalb des Kontextes zu konstruieren.

```
labelprefix=<string>
```

Diese Option gilt nur für numerische Zitierungen/Bibliografiestile und erfordert, dass die `defernumbers`-Option aus § 3.1.2.1 global aktiviert wird. Das Setzen dieser Option wird `implicit resetnumbers` für jede `\printbibliography` im Skopus des Kontexts aktiviert. Die Option ordnet den `<string>` als ein Präfix zu allen Einträgen im Referenzkontext. Beispielsweise, wenn der `<string>` A ist, die numerischen Labels werden ausgegeben werden als [A1], [A2], [A3], etc. Dies ist nützlich,

für unterteilte numerische Bibliografien, bei denen jede Subbibliografie ein anderes Präfix verwendet. Der  $\langle string \rangle$  ist für alle Stile im `labelprefix`-Feld aller betroffenen Einträge vorhanden. Sehen Sie § ?? für weitere Details.

Die `refcontext`-Umgebung kann nicht verschachtelt werden und BibLaTeX wird einen Fehler generieren, wenn Sie versuchen, dies zu tun.

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

Dieser Befehl ist ähnlich zur `refcontext`-Umgebung, außer dass es ein allein-stehender Befehl ist und keine Umgebung. Er beendet automatisch den vorherigen Kontextabschnitt (falls vorhanden) und beginnt sofort eine neue. Beachten Sie, dass der Kontextabschnitt mit dem letzten `\newrefcontext`-Befehl gestartet wird, er wird ihn bis zum Dokumentende verlängern.

Zu Beginn des Dokuments gibt es immer einen globalen Kontext, der globale Einstellungen für die einzelnen Kontextattribute enthält. Jetzt ein zusammenfassendes Beispielfür Referenzkontexte mit verschiedenen Settings.

```
\usepackage[sorting=nty]{biblatex}

\DeclareRefcontext{testrc}{sorting=nyt}

% Global reference context:
% sorting=nty
% sortingnamekeyscheme=global
% labelprefix=

\begin{document}

\begin{refcontext}{testrc}
% reference context:
% sorting=nyt
% sortingnamekeyscheme=global
% labelprefix=
\end{refcontext}

\begin{refcontext}[labelprefix=A]{testrc}
% reference context:
% sorting=nyt
% sortingnamekeyscheme=global
% labelprefix=A
\end{refcontext}

\begin{refcontext}[sorting=ydnt,labelprefix=A]
% reference context:
% sorting=ydnt
% sortingnamekeyscheme=global
% labelprefix=A
\end{refcontext}

\newrefcontext[labelprefix=B]
% reference context:
```

```

% sorting=nty
% sortingnamekeyscheme=global
% labelprefix=B
\endrefcontext

\newrefcontext}[sorting=ynt,labelprefix=C]{testrc}
% reference context:
% sorting=ynt
% sortingnamekeyscheme=global
% labelprefix=C
\endrefcontext

```

```

\assignrefcontextkeys [<key=value, ...>] {<keyword1,keyword2, ...>}
\assignrefcontextkeys* [<key=value, ...>] {<keyword1,keyword2, ...>}
\assignrefcontextcats [<key=value, ...>] {<category1, category2, ...>}
\assignrefcontextcats* [<key=value, ...>] {<category1, category2, ...>}
\assignrefcontextentries [<key=value, ...>] {<entrykey1, entrykey2, ...>}
\assignrefcontextentries* [<key=value, ...>] {<entrykey1, entrykey2, ...>}
\assignrefcontextentries [<key=value, ...>] {<*>}
\assignrefcontextentries* [<key=value, ...>] {<*>}

```

Diese Befehle setzen Zitate automatisch in die Refkontexte. Anstatt Zitierungsbefehlen in `refcontext`-Umgebung einzusetzen, die fehleranfällig sein könnten und langweilig, können Sie eine Komma-separierte Liste von `<keywords>`, `<categories>` oder `<entrykeys>` registrieren, die erzeugen Einträge mit einem der angegebenen Schlüsselwörter, Einträge in einer der in § 3.12.4 angegebenen Kategorien oder Einträge mit einer der spezifizierten Zitiertasten in einem bestimmten Refkontext durch die angegebenen `<refcontext key/values>`, die genau analysiert werden als `refcontext`-Optionen. Solche automatische `refcontext`-Zuordnungen sind spezifisch zur daktuellen Refsection. Dies ist nützlich, wenn Sie mehrere Bibliografien haben, anders sortierend als bei der Benutzung von `defernumbers`, dann muss BibLaTeX wissen, welche Bibliografie (die immer in einem bestimmt Refkontext ist) die Zitierung erzeugen soll, da die Zitate auch unterschiedliche Nummern in verschiedenen Bibliografien haben können. Sie können die gleichen Zitatschlüssel in diesen Befehlen angeben, sollten sich aber bewußt sein, das die Zuordnung in der Reihenfolge `<keywords>`, `<categories>`, `<entrykeys>` später Spezifikationen von früheren überschreiben wird. Ein Beispiel:

```

\assignrefcontextentries[labelprefix=A]{key2}
\cite{key1}
\begin{refcontext}[labelprefix=B]
\cite{key2}
\end{refcontext}

```

Hier werden die Daten für die Zitierung `key2` entnommen vom Refkontext `labelprefix=A` und nicht von `labelprefix=B` (was zu einem Label mit dem Präfix ‚A‘ und nicht ‚B‘ führt). Die gesternte Version überschreibt nicht einen lokalen Refkontext und somit:

```

\assignrefcontextentries*[labelprefix=A]{key2}
\cite{key1}
\begin{refcontext}[labelprefix=B]
\cite{key2}
\end{refcontext}

```

Die Daten für die Zitierung `key2` werden aus dem Refkontext `labelprefix=B` entnommen. Beachten Sie, dass diese Befehle nur selten benötigt werden, dann wenn Sie mehrere Bibliografien haben, in denen die gleichen Zitate sein können und BibLaTeX standardmäßig nicht sagen kann, auf welche Bibliografie sich ein Zitat bezieht. Sehen Sie die Beispieldatei `94-labelprefix.tex` für weitere Details an.

### 3.7.11 Dynamische Eintragssätze

Zusätzlich zu dem `@set`-Eingabetyp unterstützt BibLaTeX auch dynamische Mengeneinträge, die auf der 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{<key>}{<key1,key2,key3,...>}
```

Der  $\langle key \rangle$  ist der Eingabeschlüssel der Menge (des „sets“), 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<key>` 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. Sehen Sie § 3.12.5 für weitere Details.

## 3.8 Zitierfehler

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[\langle prenote \rangle][\langle postnote \rangle]{\langle keys \rangle}\langle punctuation \rangle
```

Wenn die `autopunct`-Paket Option aus § 3.1.2.1 freigegeben ist, wird sie nach jeder  $\langle punctuation \rangle$  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 (e. V.) für Details.

### 3.8.1 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[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
\Cite[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

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[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
\Parencite[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

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[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
\footcitetext[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

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.

### 3.8.2 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 Zitierbefehle werden von allen 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. In den verbose-Stilen wird das Label mit einer Fußnote versehen. Nachgestellte Zeichensetzung wird zwischen den Autor- oder Herausgebernamen und die Fußnotenmarke bewegt. `\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.

### 3.8.3 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 (e. V.)). Das folgende Beispiel illustriert die Syntax solcher ‚`multicite`‘-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 Zitierbefehl gegeben ist. Wenn nur ein optionales Argument für ein Argument der Liste 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/leer 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 §§ 3.8.1 and 3.8.2 entsprechen:

```
\cites (<multiprenote>) (<multi-postnote>) [<prenote>] [<postnote>] {<key>} ... [<prenote>] [<postnote>] {<key>}
\Cites (<multiprenote>) (<multi-postnote>) [<prenote>] [<postnote>] {<key>} ... [<prenote>] [<postnote>] {<key>}
```

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

```
\parencites (<multiprenote>) (<multi-postnote>) [<prenote>] [<postnote>] {<key>} ... [<prenote>] [<postnote>] {<key>}
\Parentcites (<multiprenote>) (<multi-postnote>) [<prenote>] [<postnote>] {<key>} ... [<prenote>] [<postnote>] {<key>}
```

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

```
\footcites (<multiprenote>) (<multi-postnote>) [<prenote>] [<postnote>] {<key>} ... [<prenote>] [<postnote>] {<key>}
\footcitetexts (<multiprenote>) (<multi-postnote>) [<prenote>] [<postnote>] {<key>} ... [<prenote>] [<postnote>] {<key>}
```

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

```
\smarcites (<multiprenote>) (<multi-postnote>) [prenote] [postnote] {<key>} . . . [prenote] [postnote] {<key>}
\Smartcites (<multiprenote>) (<multi-postnote>) [prenote] [postnote] {<key>} . . . [prenote] [postnote] {<key>}
```

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

```
\textcites (<multiprenote>) (<multi-postnote>) [prenote] [postnote] {<key>} . . . [prenote] [postnote] {<key>}
\Textcites (<multiprenote>) (<multi-postnote>) [prenote] [postnote] {<key>} . . . [prenote] [postnote] {<key>}
```

Ist die entsprechende Version multipler Literaturangaben von `\textcite` und `\Textcite`.

```
\supercites (<multiprenote>) (<multi-postnote>) [prenote] [postnote] {<key>} . . . [prenote] [postnote] {<key>}
```

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

### 3.8.4 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` (sehen Sie § ??) 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, sehen Sie § 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 folgen, 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 (⟨multiprenote⟩) (⟨multipostnote⟩) [⟨prenote⟩] [⟨postnote⟩] {⟨key⟩} . . . [⟨prenote⟩] [⟨postnote⟩] {⟨key⟩}
\Autocites (⟨multiprenote⟩) (⟨multipostnote⟩) [⟨prenote⟩] [⟨postnote⟩] {⟨key⟩} . . . [⟨prenote⟩] [⟨postnote⟩] {⟨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.

### 3.8.5 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⟩}
\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. Die gesternte Variante erzwingt effektiv „maxcitenames“ nur für diesen Befehl zu 1, so nur den ersten Namen in der Labelnamenliste auszugeben (gefolgt möglicherweise von einem „et al“ String, wenn es mehr Namen sind). Dies ermöglicht mehr einen natürlichen Textfluss, wenn man sich in einem Papier im Singular bezieht, wenn sonst `\citeauthor` eine (natürlich im Plural) Namensliste erzeugen würde.

```
\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[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
\citeyear*[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

Dieser Befehl gibt das Jahr aus (bibfieldyear-Feld oder die Jahreskomponente von date). Die gesternte Variante beinhaltet die extrayear-Information, wenn vorhanden.

```
\citedate[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
\citedate*[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

Dieser Befehl gibt das komplette Datum aus (date oder year). Die gesternte Variante beinhaltet die extrayear-Information, wenn vorhanden.

```
\citeurl[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

Dieser Befehl gibt das url-Feld aus.

```
\parentext{\langle text \rangle}
```

Dieser Befehl setzt den  $\langle Text \rangle$  in kontextabhängige runde Klammern.

```
\brackettext{\langle text \rangle}
```

Dieser Befehl setzt den  $\langle Text \rangle$  in kontextabhängige eckige Klammern.

### 3.8.6 Spezielle Befehle

Die folgenden speziellen Befehle werden ebenfalls vom BibLaTeX-Kern unterstützt.

```
\nocite{\langle key \rangle}
\nocite{*}
```

Dieser Befehl gleicht dem Standard-LaTeX-Befehl `\nocite`. Er fügt den  $\langle key \rangle$  in die Bibliografie ein, ohne den Literaturverweis auszugeben. Ist der  $\langle key \rangle$  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[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

Dieser Befehl verwendet die Bibliografietreiber für den betreffenden Eingabetyp, um einen vollständigen Literaturverweis ähnlich zu dem Bibliografieeintrag zu erstellen. Deshalb ist er eher dem Bibliografiestil, denn dem Stil für Literaturverweise zuzuordnen.

```
\footfullcite[\langle prenote \rangle][\langle postnote \rangle]{\langle key \rangle}
```

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. Das Format des Inhaltsanteils wird durch die Feldformatierungsdirektive `volcitevolume` kontrolliert; Das Format des Seite/Text-Teils wird durch die Feldformatierungsdirektive `volcitepages` gesteuert (§ 4.10.4 (e. V.)). Das Trennzeichen zwischen dem Inhaltsanteil und dem Seite/Text-Teil kann mit einer Neudefinition des Makros `\volcitedelim` geändert werden (§ 4.10.1 (e. V.)).

```
\volcites(⟨multiprenote⟩)(⟨multipostnote⟩)[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
... [⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Volcites(⟨multiprenote⟩)(⟨multipostnote⟩)[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
... [⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Die „multicite“-Version von `\volcite` und `\Volcite`.

```
\pvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Pvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Gleicht `\volcite`, basiert aber auf `\parencite`.

```
\pvolcites(⟨multiprenote⟩)(⟨multipostnote⟩)[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
... [⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Pvolcites(⟨multiprenote⟩)(⟨multipostnote⟩)[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
... [⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Die „multicite“-Version von `\pvolcite` und `\Pvolcite`.

```
\fvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\ftvolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Gleicht `\volcite`, basiert aber auf `\footcite` beziehungsweise `\footcitetext`.

```
\fvolcites(⟨multiprenote⟩)(⟨multipostnote⟩)[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
... [⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Fvolcites(⟨multiprenote⟩)(⟨multipostnote⟩)[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
... [⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Die „multicite“-Version von `\fvolcite` und `\Fvolcite`.

```
\svolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
\Svolcite[⟨prenote⟩]{⟨volume⟩}[⟨page⟩]{⟨key⟩}
```

Gleicht `\volcite`, basiert aber auf `\smartcite`.

```

\svolcites (<multiprenote>) (<multi-postnote>) [<prenote>] {<volume>} [<page>] {<key>}
... [<prenote>] {<volume>} [<page>] {<key>}
\Svolcites (<multiprenote>) (<multi-postnote>) [<prenote>] {<volume>} [<page>] {<key>}
... [<prenote>] {<volume>} [<page>] {<key>}

```

Die „multicite“-Version von `\svolcite` und `\Svolcite`.

```

\tvolcite [<prenote>] {<volume>} [<page>] {<key>}
\Tvolcite [<prenote>] {<volume>} [<page>] {<key>}

```

Gleicht `\volcite`, basiert aber auf `\textcite`.

```

\tvolcites (<multiprenote>) (<multi-postnote>) [<prenote>] {<volume>} [<page>] {<key>}
... [<prenote>] {<volume>} [<page>] {<key>}
\Tvolcites (<multiprenote>) (<multi-postnote>) [<prenote>] {<volume>} [<page>] {<key>}
... [<prenote>] {<volume>} [<page>] {<key>}

```

Die „multicite“-Version von `\tvolcite` und `\Tvolcite`.

```

\avolcite [<prenote>] {<volume>} [<page>] {<key>}
\Avolcite [<prenote>] {<volume>} [<page>] {<key>}

```

Gleicht `\volcite`, basiert aber auf `\autocite`.

```

\avolcites (<multiprenote>) (<multi-postnote>) [<prenote>] {<volume>} [<page>] {<key>}
... [<prenote>] {<volume>} [<page>] {<key>}
\Avolcites (<multiprenote>) (<multi-postnote>) [<prenote>] {<volume>} [<page>] {<key>}
... [<prenote>] {<volume>} [<page>] {<key>}

```

Die „multicite“-Version von `\avolcite` und `\Avolcite`.

```

\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.

### 3.8.7 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 (e. V.) 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 § ?? 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 § ?? 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.

### 3.8.8 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 (e. V.) 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 § ?? 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 (e. V.) 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 § ?? behandelt werden.

`\pno` Dieser Befehl erzwingt ein einzelnes Seitenpräfix im `\postnote`-Argument eines Befehls für Literaturstellen. Siehe § 3.13.3 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 § 3.13.3 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 § 3.13.3 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 § 3.13.3 für weitere Details und Verwendungsanleitungen. Das ausgegebene Suffix ist der Lokalisierungsstring `sequens`, siehe § ?. 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 § 3.13.3 für weitere Details und Anwendungsbeschreibungen. Das ausgegebene Suffix ist der Lokalisierungsstring `sequentes`, sehen Sie § ?. 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.

### 3.8.9 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`, etc.). 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-Pakets übereinzustimmen.

### 3.8.10 mcite-artige Zitierungsbefehle

Die `mcite`-Paketoption 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 §§ 3.8.1 and 3.8.2 bereit. Siehe Tabelle 9 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:

```
\mcite*[pre][post]{setA, *keyA1, *keyA2, *keyA3}
```

Er führt aus:

```
\defbibentryset{setA}{keyA1, keyA2, keyA3}%
\cite*[pre][post]{setA}
```

| 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 8: mcite-artige Befehle**

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/mcitemaplus` 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 der `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. Sehen Sie Tabelle 9 für einige Beispiele. Eine erneute Angabe der originalen Definition ist nicht nötig, aber zulässig. Im Gegensatz zu `mcitemaplus` ist hier eine teilweise Neuangabe der ursprünglichen Definition unzulässig. Man verwendet stattdessen den Eingabeschlüssel der Menge.

### 3.9 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 2 auf Seite 28. Die `<definitions>` sind `<key>= <value>`-Paare, welche einem Ausdruck einen Identifikator zuordnen:

| Eingabe                                             | Ausgabe | Kommentierung                                   |
|-----------------------------------------------------|---------|-------------------------------------------------|
| <code>\mcite{set1,*glashow,*salam,*weinberg}</code> |         | 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> |         | Redundant, aber permisible                      |
| <code>\mcite{glashow}</code>                        | [1a]    | Zitieren ein Satzmitglied                       |
| <code>\cite{weinberg}</code>                        | [1c]    | Ebenso reguläres <code>\cite</code> -Arbeiten   |

**Tabelle 9: `mcite`-artige Syntax (Beispiel der Ausgabe mit `style=numeric`- und `subentry`-Option)**

```
\DefineBibliographyStrings{american}{%
 bibliography = {Bibliography},
 shorthands = {Abbreviations},
 editor = {editor},
 editors = {editors},
}
```

Eine vollständige Liste aller Schlüssel, die von der Standardeinstellung bereitgestellt werden, findet sich in § ?? . 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. Sehen Sie § ?? 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 § 3.10.3.

```
\UndefineBibliographyExtras{<language>}{<code>}
```

Dieser Befehl wird verwendet, um die Originaldefinition aller durch `\DefineBibliographyExtras` modifizierten Befehle wieder herzustellen. Wenn ein neu definierter Befehl aus § 3.10.3 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-Kopfe 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 (e. V.) aufgelistet werden, sind durch die Standardeinstellung definiert.

### 3.10 Formatierungsbefehle

Die Befehle und Möglichkeiten, die in diesem Abschnitt erläutert werden, können zur Abstimmung der Formate von Literaturverweisen und Bibliografien verwendet werden.

#### 3.10.1 Generelle Befehle und Hooks

Die Befehle in diesem Abschnitt können mit `\renewcommand` in der Dokumentpräambel neu definiert werden. Diejenigen, die als ‚Context Sensitive‘ im Rand markiert sind, kann man auch mit `\DeclareDelimFormat` individualisieren und ausgeben mit `\printdelim` (§ 3.10.2). Man beachte, dass alle Befehle, die mit `\mk...` beginnen, ein Argument benötigen. All diese Befehle sind in `biblatex.def` definiert.

`\bibsetup` Beliebiger Code, der zu Beginn der Bibliografie ausgeführt wird, für Befehle gedacht, die das Layout der Bibliografie betreffen.

`\bibfont` Beliebiger Code, der die in der Bibliografie verwendete Schriftart bestimmt. Dieser ist sehr ähnlich zu `\bibsetup`, ist aber für wechselnde Schriftarten gedacht.

`\citesetup` Beliebiger Code, der zu Beginn eines jeden Befehls für Literaturverweise ausgeführt wird.

`\newblockpunct` Das Gliederungszeichen zwischen Bereichen unterschiedlicher Bedeutung, welches in § 4.7.1 (e. V.) erklärt wird. Die Standarddefinition wird von der Paketoption `block` kontrolliert (sehen Sie § 3.1.2.1).

`\newunitpunct` Das Gliederungszeichen zwischen Sinneinheiten, welches in § 4.7.1 (e. V.) 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 Bibliografieeintrags gesetzt wird, normalerweise ein Punkt. Die Standarddefinition ist ein Punkt.

`\entrysetpunct` Die Interpunktion, die zwischen den Bibliografieuntereinträgen eines Eintragsatzes ausgegeben wird. Die Standarddefinition ist ein Semikolon und ein Leerzeichen.

`\bibnamedelima` Dieses Trennzeichen steuert den Abstand zwischen den Elementen, die ein Namensteil sind. Es wird automatisch nach dem Vornamen eingefügt, wenn das Element länger als drei Zeichen lang ist und vor dem letzten Element ist. Die Standarddefinition ist ein wortinterner Raum, penalisiert vom Wert des `highnamepenalty`-Zählers (§ 3.10.4). Sehen Sie bitte in § 3.13.4 für weitere Details.

- `\bibnamedelimb` Dieses Trennzeichen ist zwischen dem eingesetzten Element, das einen Namensteil bildet, der `\bibnamedelima` nicht verwendet. Die Standarddefinition ist ein wortinterner Raum, penalisiert vom Wert des `lownamepenalty`-Zählers (§ 3.10.4). Sehen Sie bitte in § 3.13.4 für weitere Details.
- `\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, penalisiert vom Wert des `highnamepenalty`-Zählers (§ 3.10.4). Sehen Sie bitte in § 3.13.4 für weitere Details.
- `\bibnamedelimd` Dieses Trennzeichen ist zwischen allen Namensteilen, die `\bibnamedelimc` nicht verwendet. Die Standarddefinition ist ein wortinterner Raum, penalisiert vom Wert des `lownamepenalty`-Zählers. (§ 3.10.4). Sehen Sie bitte in § 3.13.4 für weitere Details.
- `\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.
- `\bibinitperiod` Fügt die Interpunktion nach Initialen ein, es sei denn `\bibinithyphendelim` gilt. Die Standarddefinition ist ein Punkt (`\addot`). Sehen Sie bitte in § 3.13.4 für weitere Details.
- `\bibinitdelim` Der Abstand zwischen mehreren eingefügten Initialen, es sei denn `\bibinithyphendelim` gilt. Die Standarddefinition ist ein nicht zerbrechlicher Interwort-Raum. Sehen Sie bitte in § 3.13.4 für weitere Details.
- `\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 § 3.13.4 für weitere Details.
- `\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.
- `\revsdnamepunct` Die Zeichensetzung zwischen Vor- und Nachnamen, wenn Teile ausgegeben werden, wenn der Name umgekehrt wird. Hier ist ein Beispiel eines Namens mit dem Standardkomma als `\revsdnamedelim` zu sehen:

Jones, Edward

Dieser Befehl sollte mit `\bibnamedelimd` als „reversed-name separator“ in Formatierungsanweisungen für Namenslisten verwendet werden. Bitte beachten Sie, § 3.13.4 für weitere Details.

`\bibnamedash` Der Strich, der als Ersatz für wiederkehrende Autoren oder Herausgeber in der Bibliografie 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 Bibliografie alphabetisch zu ordnen (`author` oder `editor`, falls das `author`-Feld nicht definiert ist). Bei den Standardstilen ersetzt dieses Trennzeichen an dieser Stelle den `\newunitpunct`. Die Standarddefinition ist `\newunitpunct`, das heißt, es wird nichts 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 nichts 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>24</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 Goossens, 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. Sehen Sie ebenfalls

`\finalandcomma` in § 3.10.3.

---

<sup>24</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`.

`\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.

`\andothersdelim` Das Trennzeichen, welches vor dem Lokalisationsstring ‚andothers‘ 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. Sehen Sie `\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. Sehen Sie `\finalnamedelim` für weitere Erklärungen.

`\andmoredelim` Das Trennzeichen, welches vor dem Lokalisierungsstring ‚andmore‘ 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.

`\compcedelim` Gleicht `\multicitedelim`, wird aber von mehreren Stilen für Literaturangaben zum ‚komprimieren‘ von multiplen Literaturverweisen verwendet.

`\datecircadelim`

Kontext sensitiv

Wenn die Datenformatierung mit der globalen Option `datecirca` aktiviert ist, wird das Trennzeichen nach dem lokalen ‚circa‘-Term ausgegeben. Standards beim Wortzwischenraum.

`\dateeradelim`

Kontext sensitiv

Wenn die Datenformatierung mit der globalen Option `dateera` aktiviert ist, wird das Trennzeichen vor dem lokalen `era`-Term ausgegeben. Standards beim Wortzwischenraum.

`\dateuncertainprint` Gibt Datenunsicherheitsinformationen aus, wenn die globale Option `dateuncertain` aktiviert ist. Standardmäßig gibt es die sprachenspezifische Zeichenfolge `\bibdateuncertain` aus (§ 3.10.3).

`\datecircaprint` Gibt Datenunsicherheitsinformationen aus, wenn die globale Option `datecirca` aktiviert ist. Standardmäßig gibt es den ‚circa‘ lockalen Term aus und den `datecircadelim` Trenner.

`\dateeraprint` `yearfield` gibt die Ära-Information aus, wenn die globale Option `dateera` auf ‚secular‘ oder ‚christian‘ gesetzt ist. Standardmäßig wird der `dateeradelim`-Trenner und der der entsprechend lokalisierte Ärebegriff ausgegeben (§ ??). Wenn die `dateeraauto`-Option gesetzt ist, dann wird das `\yearfield` getestet (das ist der Name eines Jahrfeldes, sowas wie ‚year‘, ‚origyear‘, ‚endeventyear‘ etc.), um zu sehen, ob sein Wert früher als die `dateeraauto`-Schwelle ist und wenn ja, dann wird die BCE/CE-Lokalisierung geschlossen sein. Die Standardeinstellung für `dateeraauto` ist 0 und so sind nur BCE/BC-Localisationsstrings Ausgabekandidaten. Ist ermittelt, ob der Beginn oder das Ende eine Ärainformation auszugeben ist, wird der `\yearfield`-Name übergeben.

`\textcitedelim` Ähnlich zu `\multicitedelim`, aber mit `\textcite` und zugehörigen Befehlen zu nehmen (§ 3.8.2). Der Standardwert ist ein Komma und ein Wortzwischenraum. Die Standardstile modifizieren diese vorläufige definition, um sicherzustellen, dass das Trennzeichen vor der letzten Zitierung der lokalisierte Term ‚and‘ ist, getrennt von Wortzwischenräumen. Sehen Sie dazu auch `\finalandcomma` und `\finalandsemicolon` in § 3.10.3.

`\nametitledelim`

Kontext sensitiv

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.

`\nameyeardelim`

Kontext sensitiv

Das Trennzeichen, welches zwischen Autor/Herausgeber und dem Jahr aus Autor-Jahr-Stilen für Literaturverweise gesetzt wird. Die Standardversion ist ein Leerzeichen.

`\namelabeldelim` Das Trennzeichen, welches zwischen dem Nemen/Title und dem Label von alphabetischen und numerischen Zitierstilen gesetzt wird. Die Standardversion ist ein Leerzeichen.

Kontext sensitiv

`\nonameyeardelim` Das Trennzeichen, welches zwischen dem Ersatz für den Labelnamen gesetzt wird, wenn er nicht vorhanden ist (in der Regel Label oder Titel in Standardstilen) und das Jahr in den Autor-Jahr-Zitierstilen. Dies wird nur verwendet, wenn kein Label da ist, jedoch ein Labelname existiert, `\nameyeardelim` wird verwendet. Die Standardversion ist ein Leerzeichen.

Kontext sensitiv

`\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:

```
\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. Sehen Sie § 3.8 für Details. Die Voreinstellung ist ein Leerzeichen.

`\postnotedelim` Das Trennzeichen, welches nach dem `\postnote`-Argument eines Literaturstellen-Befehls ausgegeben wird. Sehen Sie § 3.8 für Details. Die Voreinstellung ist ein Komma mit einem Leerzeichen.

`\mkbibname, namepart` `{\langle text \rangle}` Dieser Befehl, der ein Argument nimmt, wird verwendet, um den Namensteil `,namepart'` von Namenslistenfeldern zu formatieren. Das Standarddatenmodell definiert die Namensteile `,family'`, `,given'`, `,prefix'` und `,suffix'` und deshalb sind die folgenden Makros automatisch definiert:

```
\makebibnamefamily
\makebibnamegiven
\makebibnameprefix
\makebibnamesuffix
```

Für die Rückwärtskompatibilität mit den ererbten BibTeX-Namensteilen sind die folgenden auch definiert:

```
\makebibnamelast
\makebibnamefirst
\makebibnameaffix
```

`\relatedpunct` Der Separator zwischen dem `relatedtype`-Bibliografie-Lokalisierungsstring und den Daten aus dem ersten diesbezüglichen Eintrag. Hier ist ein Beispiel, mit einem `\relatedpunct` in einen Gedankenstrich gesetzt:

```
A. Smith. Title. 2000, (Orig. pub. as-Origtitle)
```

`\relateddelim` Der Separator zwischen den Daten von mehreren verwandten Einträgen. Die Standarddefinition ist ein optionaler Punkt plus Zeilenumbruch. Hier ist ein Beispiel, in dem die Bände A-E verbundene Einträge von 5 Bänden der Hauptarbeit sind:

```
Donald E. Knuth. Computers & Typesetting. 5 vols.
 ↪ Reading, Mass.: Addison-
Wesley, 1984-1986.
```

```
Vol. A: The TEXbook. 1984.
Vol. B: TEX: The Program. 1986.
Vol. C: The METAFONTbook. By. 1986.
Vol. D: METAFONT: The Program. 1986.
Vol. E: Computer Modern Typefaces. 1986.
```

### 3.10.2 Kontext-sensitive Trennzeichen

Die Trennzeichen, beschrieben in § 3.10.1, sind global definiert. Das heißt, unabhängig davon, wo Sie diese verwenden, sie geben die gleiche Sache aus. Dies ist für Begrenzer nicht unbedingt wünschenswert, die Sie für verschiedene Dinge in unterschiedlichen Kontexten ausgeben möchten. Hier bedeutet ‚context‘, Dinge wie like ‚inside a text citation‘ oder ‚inside a bibliography item‘. Aus diesem Grund bietet BibLaTeX eine anspruchsvollere Begrenzerspezifikation und eine Benutzeroberfläche neben dem Standard, basierend auf den normalen Makros, mit `\newcommand` definiert.

```
\DeclareDelimFormat[⟨context, ...⟩]{⟨name, ...⟩}{⟨code⟩}
\DeclareDelimFormat* [⟨context, ...⟩]{⟨name, ...⟩}{⟨code⟩}
```

Deklariert die Begrenzermakros in Komma-separierten Namenslisten ( $\langle names \rangle$ ) mit dem „replacement test“  $\langle code \rangle$ . Wenn die optionale Komma-separierte Kontextliste ( $\langle contexts \rangle$ ) gegeben ist, deklarieren Sie die Namen ( $\langle names \rangle$ ) nur für diese Kontexte.  $\langle names \rangle$  ohne diese  $\langle contexts \rangle$  verhalten sich genauso wie die globalen Begrenzungsdefinitionen, mit `\newcommand` gegeben – nur ein einfaches Makro mit einem Ersatz/replacement, das als `\name` verwendet werden kann. Sie können jedoch auch die auf diese Weise erzeugten Begrenzermakros aufrufen, indem Sie `\printdelim` nehmen, welches kontextbewusst ist. Die gesternte Version löscht alle  $\text{kontext}(\langle context \rangle)$ -spezifischen Deklarationen für alle Namen ( $\langle names \rangle$ ) zuerst.

```
\printdelim[⟨context⟩]{⟨name⟩}
```

Gibt ein Trennzeichen mit dem Namen  $\langle name \rangle$  aus, lokal einen optionalen Kontext ( $\langle context \rangle$ ) zuerst festlegend. Ohne optional  $\langle context \rangle$ , `\printdelim` nimmt es den aktuell aktiven Begrenzerkontext.

Trennzeichenkontexte sind einfach ein String, der Wert des internen Makros `\blx@delimcontext`, welcher manuell mit dem Befehl `\delimcontext` eingestellt werden kann.

```
\delimcontext{⟨context⟩}
```

Setzt den Begrenzerkontext ein ( $\langle context \rangle$ ). Diese Einstellung ist nicht global, so dass Trennzeichenkontexte verschachtelt werden können, nehmend dafür die üblichen LaTeX-Gruppen-Verfahren.

```
\DeclareDelimcontextAlias{⟨alias⟩}{⟨name⟩}
```

Das kontext-sensitive Trennsystem schafft Begrenzerkontexte, die auf dem Namen der Zitierbefehle basieren (‚parencite‘, ‚textcite‘ etc.), übergebend dem `\DeclareCiteCommand`. In bestimmten Fällen, in denen geschachtelte Definitionen von Zitierbefehlen, wobei der Befehl `\DeclareCiteCommand` sich selbst aufruft (sehen Sie die Definition von `\textcite` in `author-year-icompile` an beispielsweise). Der Begrenzerkontext ist dann in der Regel inkorrekt und die Begrenzerspezifikation funktioniert nicht. Beispielsweise, die Definition von `\textcite`

ist tatsächlich definiert und verwendet `\cbx@textcite`, und so ist der Kontext automatisch auf `cbx@textcite` gesetzt, wenn die Zitierung ausgegeben wird. Begrenzerdefinitionen erwarten den Kontext `textcite` zu sehen, deshalb funktionieren sie nicht. Deshalb wird dieser Befehl für Stilautoren zur Verfügung gestellt, für Aliase der Kontext `\alias` zum Kontext `\name`. Zum Beispiel:

```
\DeclareDelimcontextAlias{cbx@textcite}{textcite}
```

Das (dies ist die Standardeinstellung) stellt es sicher, wenn innerhalb des Zitierbefehls `\cbx@textcite` der Kontext tatsächlich, wie erwartet, `textcite` ist.

BibLaTeX hat mehrere Standardkontexte, welche automatisch an verschiedenen Stellen festgelegt sind:

**none** Zu Dokumentbeginn.

**bib** Innerhalb einer Bibliografie, begonnen mit `\printbibliography` oder in einem `\usedriver`.

**biblist** Innerhalb eine Bibliografieliste, begonnen mit `\printbiblist`.

**,citecommand'** Innerhalb eines Zitierbefehls `\citecommand`, festgelegt mit einem `\DeclareCiteCommand`.

Beispielsweise sind die Standards für `\nametitledelim`:

```
\DeclareDelimFormat{nametitledelim}{\addcomma\space}
\DeclareDelimFormat[textcite]{nametitledelim}{\addspace}
↔ }
```

Dies bedeutet, dass `\nametitledelim` global als `,\addcomma\space'` definiert ist, so wie per Standardtrennzeicheninterface. Jedoch kann zusätzlich das Trennzeichen mit `\printdelim` ausgegeben werden, das das Gleiche wie `\nametitledelim` ausgeben würde, außer in einem `\textcite`, indem es `\addspace` ausgeben würde, das mehr geeignet ist, für die Textkompilierung. Falls gewünscht, kann ein Zusammenhang mit dem optionalen Argument mit `\printdelim` erzwungen werden, so

```
\printdelim[textcite]{nametitledelim}
```

Würde `\addspace` unabhängig vom Kontext des `\printdelim` ausgeben. Kontexte sind nur willkürliche Zeichenfolgen und so können Sie welche jederzeit einführen, nehmend `\delimcontext`. Wenn `\printdelim` für den Begrenzer (`\name`) keinen besonderen Wert im aktuellen Kontext findet, ist es einfach `\name` auszugeben. Dies bedeutet, dass Stilautoren `\printdelim` nehmen können und Benutzer können erwarten, mit `\renewcommand` Trennzeichen definieren zu können, mit einer Einschränkung – eine solche Definition wird keine Trennzeichen ändern, die sind festgelegt.

```
\DeclareDelimFormat{delima}{X}
\DeclareDelimFormat[textcite]{delima}{Y}
\renewcommand*{delima}{Z}
```

Hier, `\delima` druckt immer ‚Z‘. `\printdelim{delima}` in jedem Kontext, anders als ‚textcite‘, das auch `\delima` und hence ‚Z‘ ausgibt, aber in einem ‚textcite‘-Kontext ‚Y‘ ausgibt. Sehen Sie die `04-delimiters.tex`-Beispieldatei, die mit BibLaTeX arbeitet, für weitere Informationen.

### 3.10.3 Sprachspezifische Befehle

Die Befehle in diesem Abschnitt sind sprachspezifisch. Wenn man sie neu definiert, muss man die neue Definition in einen `\DeclareBibliographyExtras`-Befehl einbetten (in eine `.lbr`-Datei) oder einen `\DefineBibliographyExtras`-Befehl (Nutzerdokumente), sehen Sie § 3.9 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. Standards sind zu `\textendash`.
- `\bibrangesep` Der sprachspezifische Separator, der zwischen mehreren Bereichen verwendet wird. Standard: Ein Komma folgt einem Leerraum.
- `\bibdatedash` Der sprachspezifische Bindestrich, der für die Abgrenzung von Datumsangaben verwendet wird. Standards sind zu `\hyphen`.
- `\bibdaterangesep` Der sprachspezifische Separator, verwendbar bei Datumsbereichen. Standard zu `\textendash` für alle Datumsformate, außer `ymd`, die standardmäßig sind mit einem `\slash`. Die `Datumsformatoption edtf` ist hartcodiert auf `\slash`, da dies ein standardkonformes Format ist.
- `\mkbibdatelong` Nimmt die Namen der drei Felder als Argumente, die sich auf die drei Datumskomponenten beziehen (in der Reihenfolge Jahr/Monat/Tag) und seine Wirkung dafür verwendet, das Datum im sprachspezifischen ausgeschriebenen Format auszugeben.
- `\mkbibdateshort` Gleicht `\mkbibdatelong`, verwendet aber die kurze Version des sprachspezifischen Datumsformats.
- `\mkbibtimezone` Modifiziert einen Zeitonenstring, als ein einzelnes Argument übergeben. Standardmäßig ändert dies ‚Z‘ auf den Wert von `\bibtimezone`.
- `\bibdateuncertain` Die sprachspezifischen Marker zur Verwendung nach unsicheren Daten, wenn die globale Option `dateuncertain` aktiviert ist. Standardmäßig ein Leerraum, gefolgt von einem Fragezeichen.
- `\bibdateeraprefix` Ein sprachspezifischer Marker, der als ein Präfix zu Beginn von BCE/BC-Daten, in einem Datumsbereich, wenn die Option `dateera` auf ‚astronomical‘ gesetzt ist. Standardmäßig auf `\textminus`, wenn definiert und `\textendash`.
- `\bibdateeraendprefix` Ein sprachspezifischer Marker, der als ein Präfix am Ende von BCE/BC-Daten, in einem Datumsbereich, wenn die Option `dateera` auf ‚astronomical‘ gesetzt ist. Standardmäßig ein kleiner Leerraum (thin space), gefolgt von `\bibdateeraprefix` wenn `\bibdaterangesep` auf eine Bindestrich (dash) und auf `\bibdateeraprefix` gesetzt ist. Dies ist ein separates Makro, so dass Sie mehr Raum vor einen negativen Datumsmarker hinzufügen können, beispielsweise folgt ein Bindestrich einem Datumsbereichsmarker, so kann dies ein wenig seltsam aussehen.

- `\bibtimesep` Der sprachspezifische Marker, der Zeitkomponenten trennt. Der Standardwert ist ein Doppelpunkt.
- `\bibtimezonesep` Der sprachspezifische Marker, der eine optionale zeitkomponente von einer Zeit trennt. Ist standardmäßig leer.
- `\bibtzminsep` Der sprachspezifische Marker, der die Stunden- und Minutenkomponente der Offsetzeitzonen trennt. Der Standardwert ist ein `\bibtimesep`.
- `\bibdatetimesep` Der sprachspezifische Separator, der zwischen Datums- und Zeitkomponenten ausgegeben wird, ausgegeben wird, wenn Zeitkomponenten mit Datumskomponenten ausgegeben werden (sehen Sie `<datetype>dateusetime`-Option in § 3.1.2.1 an). Defaults zu einem Leerraum für nicht-EDTF-Ausgabeformate und 'T' für das EDTF-Ausgabeformat.
- `\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 Goossens, Frank Mittelbach, and Alexander
 ↪ Samarin
```

`\finalandcomma` ist das Komma vor dem Wort ‚and‘. Siehe ebenfalls `\multinamedelim`, `\finalnamedelim` und `\revsdnamedelim` in § 3.10.1.

- `\finalandsemicolon` Gibt das Semikolon entgültig aus, vor dem entgültigen ‚and‘ in einer Liste von Listen, ggf. in der jeweiligen Sprache. Hier ist ein Beispiel:

```
Goossens, Mittelbach, and Samarin; Bertram and Wenworth
 ↪ ; and Knuth
```

`\finalandsemicolon` ist das Semikolon vor dem Wort ‚and‘. Sehen Sie auch `\textcitedelim` in § 3.10.1.

`\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⟩}`

Ähnlich mit `\mkbibordinal`, ist aber für die Verwendung mit dem Begriff ‚series‘.

### 3.10.4 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 Bibliografie. Es findet nur bei Bibliografiestilen Anwendung, die Kesselsätze 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 Bibliografie. 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 Bibliografie 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 Bibliografie 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 Bibliografie 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 Bibliografie 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 Bibliografie. 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>25</sup>

<sup>25</sup>Die Standardwerte, die `abbrvpenalty`, `lownamepenalty` und `highnamepenalty` zugewiesen werden, sind absichtlich sehr niedrig, um überfüllte Zellen zu vermeiden. Das bedeutet, dass

`highnamepenalty` Dieser Zähler beinhaltet eine Strafe, welche den Zeilenumbruch bei Namen beeinflusst. In §§ 3.13.4 and 3.10.1 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` Gleich `highnamepenalty`. Erklärungen finden sich in §§ 3.13.4 and 3.10.1. 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.

### 3.10.5 Mehrzweckbefehle

Die Befehle dieses Abschnitts sind multifunktionale Befehle, die generell verfügbar sind, nicht nur für Zitierungen und die Bibliografie.

`\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 von den `babel/polyglossia`-Paketen 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 von den `babel/polyglossia`-Paketen 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 von den `babel/polyglossia`-Paketen 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 von den `babel/polyglossia`-Paketen für einige Sprachmodule zur Verfügung gestellt wird.

`\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.

---

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. 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 Bibliografie in `\raggedright` zu setzen, um suboptimale Zeilenumbrüche zu vermeiden. In diesem Fall kann auch die kleinste Standardstrafe einen sichtbaren Unterschied zeigen.

`\mknumalph{⟨integer⟩}`

Nimmt eine ganze Zahl aus dem Bereich 1–702 als Argument und wandelt sie 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 Akronym für die *small caps*-Variante der aktuellen Schrift nimmt, 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{\'e}dition sp\'eciale
```

Dies wird ‚Édition spéciale‘ oder ‚édition spéciale‘ liefern. Wenn die Zeichenfolge mit einem Befehl am beginn aktiviert wird, dann gibt es sowas wie `\ae` oder `\oe` aus, setzen Sie einfach den Befehl in das `⟨character⟩`-Argument:

```
\autocap{\oe}uvres
```

Daraus ergibt sich ‚Œuvres‘ oder ‚œuvres‘.

### 3.11 Sprachspezifische Anmerkungen

Die Möglichkeiten, die in diesem Kapitel diskutiert werden, sind spezifisch für bestimmte Lokalisierungsmodule.

#### 3.11.1 Amerikanisches Englisch

Das amerikanische Lokalisation-Modul verwendet `\uspunctuation` aus § ??, 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.

### 3.11.2 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‘

### 3.11.3 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 Bibliografie sollten gut funktionieren, vorausgesetzt man nutzt Biber als Backend, aber englische und andere Titel in der Bibliografie könnten als griechische Buchstaben dargestellt werden. Falls man Bibliografien mit mehreren Schriften braucht, ist XeLaTeX die einzige vernünftige Wahl.

#### 3.11.4 Russisch

Wie das griechische Lokalisierungsmodul benötigt das Russischmodul auch UTF-8-Unterstützung. Es wird nicht mit einer anderen Kodierung arbeiten.

### 3.12 Anwendungshinweise

Der folgende Abschnitt gibt einen grundlegenden Überblick über das BibLaTeX-Paket und zeigt einige typische Anwendungsszenarien.

#### 3.12.1 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 Bibliografie und spezifiziert den/die `bib`-Datei(en). Die Dateiendung wird weggelassen. Mit BibLaTeX werden Quellen in der Präambel mit `\addbibresource` durch die Nutzung des vollen Namens mit `.bib`-Endung spezifiziert. Die Bibliografie wird mit dem `\printbibliography`-Kommando gedruckt, welches eventuell mehrere Male benutzt wird (sehen Sie § 3.7 für Details). Der Dokumentkörper kann mehrere Zitierkommandos enthalten (§ 3.8). Die Verarbeitung dieses Beispiels erfordert eine bestimmte Prozedur, die nachfolgend gezeigt wird. Vorausgesetzt unser Beispiel heißt `example.tex` und unsere Bibliografiedaten sind in `bibfile.bib`, dann ist die Prozedur folgende:

1. Laufenlassen `latex` von `example.tex`. Falls die Datei irgendwelche Zitate enthält, enthält BibLaTeX die betreffenden Daten von jedem Zitat, BibLaTeX wird die entsprechenden Daten von Biber anfordern, indem Befehle in die `auxiliar`-Datei (Hilfsdatei) `example.bcf` eingeschrieben werden.

2. Laufenlassen von `biber` auf dem Pfad `example.bcf`. Biber wird die Daten von `bibfile.bib` holen und sie in die auxiliary-Datei `example.bbl` in einem Format, das von BibLaTeX verarbeitet werden kann, schreiben.
3. Laufenlassen `latex` von `example.tex`. BibLaTeX wird die Daten von the data aus `example.bbl` lesen und alle Zitate sowie die Bibliografie ausgeben.

### 3.12.2 Hilfsdateien

Das BibLaTeX-Paket verwendet nur eine `auxiliar-bcf`-Datei. Auch wenn Zitierungsbefehle via `\include` eingebracht werden, Sie müssen nur Biber mit der Haupt-`bcf`-Datei ausführen. Alle Informationen, die Biber braucht, sind in der `bcf`-Datei, einschließlich von Informationen über alle „refsections“, wenn mehrere `refsection`-Umgebungen sind (sehen Sie § 3.12.3).

### 3.12.3 Mehrere Bibliografien

In einer Sammlung von Artikeln von verschiedenen Autoren, wie ein Konferenzprotokoll zum Beispiel, ist es üblicher eine Bibliografie 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 Bibliografie dargestellt.

```
\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 `\printbibliographie` in einer `refsection`-Umgebung benutzt wird, beschränkt es automatisch den Bereich der Referenzliste auf die betreffende `refsection`-Umgebung. Für eine wachsende Bibliografie, 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 Bibliografietitels. Das ist die Definition, welche auf die Untertitel in der Bibliografie 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`-Befehl für jeden Teil der unterteilten Bibliografie lästig ist, bietet BibLaTeX ein Kürzel. Der `\bibbysection`-Befehl 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 Bibliografie dann folgendermaßen erzeugt werden:

```

\printbibheading
\bibbysection[heading=subbibliography]

```

Wenn man ein Format ohne eine wachsende Bibliografie 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. Das nächste Beispiel könnte ebenso in § 3.12.4 genutzt werden, da es optisch eine globale Bibliografie, in mehrere Segmente unterteilt, erzeugt.

```

\documentclass{...}
\usepackage{biblatex}
\defbibheading{subbibliography}{%
 \section*{References for Chapter \ref{refsegment:
 ↪ \therefsection\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 Referenzsegmente läuft:

```

\printbibheading
\bibbysegment[heading=subbibliography]

```

Das ist äquivalent zu je einem `\printbibliography` Kommando für jedes Segment in der gewöhnlichen `refsection`.

### 3.12.4 Unterteilte Bibliografien

Es ist sehr üblich, eine Bibliografie nach bestimmten Kriterien zu unterteilen. Zum Beispiel: Man will vielleicht gedruckte und online Quellen separat drucken oder eine Bibliografie in Primär- und Sekundärliteratur teilen. Der erstere Fall ist einfach, weil man den Eintragsstypen als Kriterium für die `type`- und `nottype`-Filter von `\printbibliography` nutzen kann. Das nächste Beispiel demonstriert außerdem, wie man angepasste Untertitel für zwei Teile der Bibliografie erzeugt.

```

\documentclass{...}
\usepackage{biblatex}
\addbibresource{...}
\begin{document}
...
\printbibheading

```

```

\printbibliography[notype=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 ausgeben, dessen Eintrags-typ @book ist und dessen keywords-Feld das Schlüsselwort ‚abc‘ aber nicht ‚xyz‘ beinhaltet. Wenn man Bibliografiefilter in Verbindung mit einem numerischen Stil verwendet, sehen Sie § 3.13.5. Falls man einen komplexen Filter mit bedingten Ausdrücken benötigt, nutzt man die `filter`-Option in Verbindung mit einem spezifizierten Filter, definiert durch `\defbibfilter`. Sehen Sie § 3.7.9 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 Bibliografie 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 Bibliografie Primär- oder Sekundärliteratur sein soll, muss man die Bibliografiefilter 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 Bibliografiekategorien. 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 Bibliografie explizit zu deklarieren, weil es einen `\bibbycategory`-Befehl gibt, welcher automatisch über alle Kategorien läuft. Er ist äquivalent zu je einem `\printbibliographie`-Befehl 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.

### 3.12.5 Eintragssätze

Ein Eintragssatz ist eine Gruppe von Einträgen, die als Einzelreferenzen zitiert wurden und als einzelne Einheit in der Bibliografie 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` (§ 3.7.11) 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 § ?? für mehr Informationen ansehen.

**3.12.5.1 Statische Eintragssätze** Statische Eintragssätze sind in der `bib`-Datei definiert, wie jeder andere Eintrag. Man definiert 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 Eintragsschlüsseln definiert:

```

@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.

**3.12.5.2 Dynamische Eintragssätze** Dynamische Eintragssätze 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 § 3.7.11 definiert werden:

```

\defbibentryset{set1}{key1, key2, key3}

```

Dynamische Eintragssätze 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.

### 3.12.6 Datencontainer

Der `@xdata`-Eintragstyp dient als Datencontainer, der eines oder mehrere Felder hält. Diese Felder können durch andere Einträge über das `xdata`-Feld vererbt werden. `@xdata`-Einträge können in der Bibliografie nicht zitiert oder hinzugefügt werden. Dieser Datenvererbungsmechanismus ist für feste `teld`-Kombinationen wie `publisher/location` und für andere häufig verwendete Daten nützlich:

```
@XData{hup,
 publisher = {Harvard University Press},
 location = {Cambridge, Mass.},
}
@Book{...,
 author = {...},
 title = {...},
 date = {...},
 xdata = {hup},
}
```

Mit Hilfe einer getrennten Liste von Schlüsseln in seinem `xdata`-Feld kann ein Eintrag Daten aus mehreren `@xdata`-Einträgen erben. Mehrstufige `@xdata`-Einträge werden ebenfalls unterstützt, d. h. ein `@xdata`-Eintrag kann auf einen oder mehrere `@xdata`-Einträge referieren:

```
@XData{macmillan:name,
 publisher = {Macmillan},
}
@XData{macmillan:place,
 location = {New York and London},
}
@XData{macmillan,
 xdata = {macmillan:name,macmillan:place},
}
@Book{...,
 author = {...},
 title = {...},
 date = {...},
 xdata = {macmillan},
}
```

Sehen Sie auch §§ 2.1.1 and 2.2.3.

### 3.12.7 Informationen zu elektronischen Publikationen

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 = {identifizier},
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 Paketooptionen `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 Bibliografie 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:  $\langle pmid \rangle$ “ statt der langen URL. Falls der hyperref-Support aktiviert ist, wird  $\langle pmid \rangle$  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>26</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 § ??, um zu sehen, wie man einen geeigneten Support für andere Eprint-Quellen hinzufügt.

### 3.12.8 Externe Inhaltsangaben und Anmerkungen

Stile, die die Felder `abstract` und/oder `annotation` ausgeben, können einen alternativen Weg unterstützen, um Inhaltsangaben oder Anmerkungen zu der Bibliografie hinzuzufügen. Statt den Text in der `bib`-Datei einzubinden, kann er ebenso in einer externen LaTeX-Datei gespeichert werden. Zum Beispiel, statt zu sagen:

```
@Article{key1,
 ...
 abstract = {This is an abstract of entry `key1
 ↪ `'.}
}
\end{lstlisting}
%
Man kann in der \file{bib}-Datei eine Datei namens \path
 ↪ {bibabstract-key1.tex} schaffen
und das Abstrakt in diese Datei setzen:

\begin{ltxexample}
This is an abstract of entry `key1'.
\endinput
```

<sup>26</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.

Der Name der externen Datei muss vor dem Eintragsschlüssel 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 Performanzgründen deaktiviert. Man beachte außerdem, dass jedes `abstract-` und `annotation-`Feld in der `bib`-Datei den Vorzug 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 (e. V.) für weitere Informationen anschauen.

### 3.13 Hinweise und Warnungen

Dieser Abschnitt bietet zusätzliche Nutzungshinweise und zeigt einige übliche Probleme und potentielle Missverständnisse.

#### 3.13.1 Nutzung mit KOMA-Script-Klassen

Wenn man BibLaTeX in Verbindung mit der `scrbook-`, `scrreprt-` oder `scrartcl`-Klasse benutzt, sind die Titel `bibliography` und `shorthands` von § 3.7.7 ansprechbar für die bibliografiebezogenen Optionen dieser Klassen.<sup>27</sup> Man kann die Standardtitel überschreiben, indem man die `heading`-Option von `\printbibliography`, `\printbibheading` und `\printshorthands` benutzt. Siehe §§ 3.7.2, 3.7.3, 3.7.7 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:

```
\ifkomabibtotoc{<true>}{<false>}
```

Wird zu `<true>`, falls die Klasse die Bibliografie zu dem Inhaltsverzeichnis hinzufügen würde und zu `<false>` anderenfalls.

```
\ifkomabibtotocnumbered{<true>}{<false>}
```

Wird zu `<true>`, falls die Klasse die Bibliografie zu dem Inhaltsverzeichnis als nummerierten Abschnitt hinzufügen würde, anderenfalls wird es zu `<false>`. Falls dieser Test `<true>` bringt, wird `\ifkomabibtotoc` ebenso immer `<true>` zurückgeben, aber nicht anderes herum.

#### 3.13.2 Nutzung mit der Memoir-Klasse

Wenn man BibLaTeX mit der `memoir`-Klasse benutzt, haben die meisten Möglichkeiten für die Adaption der Bibliografie keinen Effekt. Man benutze stattdessen die dementsprechenden Hilfen von dem Paket (§§ 3.7.2, 3.7.7, 3.7.8). Statt

<sup>27</sup>Dies bezieht sich auf die traditionelle Syntax dieser Optionen (`bibtotoc` und `bibtotocnumbered`) und ebenso auf die `<key>=<value>`-Syntax, die in KOMA-Script 3.x hinzugefügt wurde, z.B. auf `bibliography=nottotoc`, `bibliography=totoc` und `bibliography=totocnumbered`. 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.

`\bibsection` von `memoir` umzudefinieren, nutze man die `heading`-Option von `\printbibliography` und `\defbibheading` (§§ 3.7.2 and 3.7.7). Statt `\prebibhook` und `\postbibhook` nutze man die `prenote`- und `postnote`-Option von `\printbibliography` und `\defbibnote` (§§ 3.7.2 and 3.7.8). Alle Standardtitel werden beim Laden derart adaptiert, dass sie sich mit dem standardmäßigen Layout dieser Klasse vermischen. Die Defaulttitel `bibliography` und `shorthands` (§ 3.7.7) 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 (§ 3.10.4). Dieser Abschnitt stellt ebenfalls einige zusätzliche Längenregister vor, welche `\biblistextra` von `memoir` entsprechen. `\setbiblabel` bildet nicht auf eine einzelne Einheit des BibLaTeX-Pakets ab, da der Stil aller Beschriftungen von dem Bibliografiestil kontrolliert wird. Sehen Sie § ?? 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:

```
\ifmemoirbibintoc{<true>}{<false>}
```

Wird zu `<true>` oder `<false>`, abhängig von den `\bibintoc`- und `\nobibintoc`-Schaltern von `memoir`. Dies ist ein LaTeX-rontend für den `\ifnobibintoc`-Test von `memoir`. Man beachte, dass die Logik des Test umgekehrt ist.

### 3.13.3 Seitenzahlen in Zitaten

Falls das `<postnote>`-Argument für den Zitierbefehl 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 betrachten, welche mit ‚p.‘ oder einem anderen String präfixt wird, abhängig von dem `pagination`-Feld (sehen Sie § 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 werden:

```
\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 § 3.8.8 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 einen `\ppno`-Befehl, welcher ein Bereichspräfix erzwingt, und ebenso einen `\nopp`-Befehl, welcher Präfixe unterbindet:

```
\cite[\ppno~27a--28c]{key}
\cite[\nopp 25]{key}
```

Diese Befehle 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~sqq.]{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, sehen Sie § 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  $\langle postnote \rangle$ -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. Sehen Sie §§ 4.3.2 und 4.4.2 (e. V.) in der Autorenanleitung für weitere Details.

### 3.13.4 Namenteile und Namenszwischenräume

Das BibLaTeX-Paket bietet den Nutzern und Stilautoren eine sehr gute „gekörnte“ Kontrolle der Namenszwischenräume und des Zeilenumbruchs. Die folgenden diskutierten Befehle sind in §§ 3.10.1 and ?? 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 Namenteil kann ein einzelner Name oder aus mehreren Teilen zusammengesetzt sein. Beispielsweise, der Namenteil ‚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:

|                                |                                                                                                                                                                                                                     |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>a=\bibnamedelima</code>  | 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.                                        |
| <code>b=\bibnamedelimb</code>  | Vom Backend zwischen alle Namenteilelemente eingefügt, wo <code>\bibnamedelima</code> nicht anwendbar ist.                                                                                                          |
| <code>c=\bibnamedelimc</code>  | Eingefügt von einer <i>Formatting directive</i> zwischen dem Namenspräfix und dem Nachnamen, wenn <code>useprefix=true</code> . Wenn <code>useprefix=false</code> , dann ist <code>\bibnamedelimd</code> zu nehmen. |
| <code>d=\bibnamedelimd</code>  | Eingefügt von einer <i>Formatting directive</i> zwischen Namensteilen, wo <code>\bibnamedelimc</code> nicht anwendbar ist.                                                                                          |
| <code>i=\bibnamedelimi</code>  | Ersetzt <code>\bibnamedelima/b</code> nach Initialen (großen Anfangsbuchstaben).                                                                                                                                    |
| <code>p=\revsdnamepunct</code> | Inserted by a formatting directive after the family name when the name parts are reversed.                                                                                                                          |

Nun, wie Trennzeichen verwendet werden:

John|<sub>a</sub> Edward|<sub>d</sub> Doe  
Doe|<sub>p</sub>|<sub>d</sub> John|<sub>a</sub> Edward

Initialen erhalten in der bib-Datei ein spezielles Trennzeichen:

J.|<sub>i</sub> Edward|<sub>d</sub> Doe

Nun wollen wir komplexere Namen betrachten: „Charles-Jean Étienne Gustave Nicolas de La Vallée Poussin“. Diese Namen setzt sich aus folgenden Teilen zusammen:

Given Charles-Jean Étienne Gustave Nicolas  
Prefix de  
Family 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 das Backend eingefügt wurde. Das Backend behandelt die Namenteile und kümmert sich um die Trennzeichen zwischen den Elementen, dies baut einen Namenteil um, jeden Namenteil 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=\bibinithyphndelim</code> | Eingefügt vom Backend zwischen den Initialen eines Bindestrich-Namens, ersetzend <code>\bibinitperiod</code> und <code>\bibinitdelim</code> . |

So werden sie eingesetzt:

J|<sub>a</sub>.|<sub>b</sub> E|<sub>a</sub>. Doe  
K|<sub>c</sub>.-H|<sub>a</sub>. Mustermann

### 3.13.5 Bibliografiefilter und Zitatbeschriftungen

Die Zitatbeschriftungen, die mit diesem Paket generiert werden, werden in die Liste der Referenzen übertragen, bevor sie von irgendeinem Bibliografiefilter geteilt wird. Sie sind garantiert einzigartig bezüglich des gesamten Dokuments (oder einer `refsection`-Umgebung), egal wie viele Bibliografiefilter genutzt werden. Wenn man ein numerisches Zitateschema benutzt, wird dies meistens zu einer diskontinuierlichen Nummerierung in geteilten Bibliografien 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 Bibliografie ausgegeben wird.

### 3.13.6 Aktive Zeichen in Bibliografietiteln

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 Befehle, sondern werden einfach wörtlich gedruckt. Die direkteste Lösung besteht darin `\defbibheading` hinter `\begin{document}` zu verschieben. Alternativ kann man die `\shorthandon` und `\shorhandoff` Befehle von `babel` nutzen, um die Kürzel temporär in der Präambel zu aktivieren. Das oben bezieht sich ebenfalls auf Bibliografieanmerkungen und das `\defbibnote`-Befehle.

### 3.13.7 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 hinzugefügt (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 Befehle `\newrefsection` und `\newrefsegment` keine Gruppe erzeugen. Sehen Sie §§ 3.7.4 and 3.7.5 für Details.

## 3.14 Anwenden des alternativen BibTeX-Backends

Um alle hier beschriebenen Funktionen zu nutzen, muss BibLaTeX mit dem Biber-Programm als Backend benutzt werden. Tatsächlich setzt die Dokumentation dieses voraus. Doch für eine *begrenzte* Anzahl von Anwendungsfällen ist es auch möglich, das schon lange etablierte BibTeX-Programm zu nehmen, entweder das 7-bit `bibtex` oder 8-bit `bibtex8` als das unterstützende Backend. Dies arbeitet in Vielem wie Biber, nur mit der Einschränkung dass BibTeX viel mehr als Backend beschränkt ist.

Das Verwenden von BibTeX als Backend erfordert, dass die Option `backend=bibtex` oder `backend=bibtex8` zum Laden gegeben wird.

Das BibLaTeX-Paket wird dann die entsprechenden Daten für die Verwendung mit BibTeX in die Hilfsdatei(en) schreiben und eine spezielle Datendatei (automatisch eingeschlossen zum Lesen von BibTeX). Das BibTeX (8)-Programm sollte dann für jede Hilfsdatei ausgeführt werden: BibLaTeX wird alle erforderlichen Dateien in der log-Datei auflisten.

Schlüsseleinschränkungen vom BibTeX-Backend sind:

- Die Sortierung ist global und limitiert von Ascii.
- Keine erneute Codierung ist möglich und somit müssen die Datenbankeinträge in der LICR-Form sein, um zuverlässig arbeiten zu können.
- Das Datenmodell ist fest.
- Querverweisen ist begrenzter und die Eintragssätze müssen in die `.bib`-Datei geschrieben werden.
- Feste Speicherkapazität (nehmend die `--wolfgang`-Option mit `bibtex8` ist streng empfohlen, um die Wahrscheinlichkeit eine Debatte dazu hier zu minimieren).

## 4 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 das 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.

### 3.7 2016-12-08

Corrected default for `\bibdateeraprefix` . . . . . ??  
Added `\DeclareSortInclusion` . . . . . ??  
Added `\relateddelim<relatedtype>` . . . . . 3.10.1

### 3.6 2016-09-15

Corrected some documentation and fixed a bug with `labeldate` localisation strings.

### 3.5 2016-09-10

Added `\ifuniquebaretitle test` . . . . . ??  
Documented `\labelnamesource` and `\labeltitlesource` . . . . . ??  
Added `\bibdaterangesep` . . . . . 3.10.3  
Added `refsection` option to `\DeclareSourceMap` . . . . . ??  
Added `suppress` option to inheritance specifications . . . . . ??  
Added `\ifuniquework` . . . . . ??  
Changed `\DeclareStyleSourceMap` so that it can be used multiple times  
??  
Added `\forcezerosy` and `\forcezerosmdt` . . . . . ??  
Changed `\mkdatezeros` to `\mkyearzeros`, `\mkmonthszeros` and  
`\mkdayzeros` . . . . . ??  
Added `namehash` and `fullhash` for all name list fields . . . . . ??  
Generalised `giveninits` option to all nameparts . . . . . 3.1.2.3  
Added `inits` option to `\DeclareSortingNamekeyScheme` . . . . . ??  
Added `\DeclareLabelalphaNameTemplate` . . . . . ??  
Added full EDTF Levels 0 and 1 compliance for parsing and printing times 2.3.8  
Changed dates to be fully EDTF Levels 0 and 1 compliant. Associated tests and  
localisation strings . . . . . 2.3.8  
Added `timezeros` . . . . . 3.1.2.1  
Added `mktimezeros` . . . . . ??  
Changed `iso8601` to `edtf` . . . . . 3.1.2.1  
Added `\DeclareUniquenameTemplate` . . . . . ??

|                                                                                                              |         |            |
|--------------------------------------------------------------------------------------------------------------|---------|------------|
| Removed experimental RIS support                                                                             |         |            |
| sortnamekeyscheme and useprefix can be now be set per-namelist and per-name for BibTeX datasources . . . . . | ??      |            |
| Added \DeclareDelimcontextAlias . . . . .                                                                    | 3.10.2  |            |
| Added Estonian localisation (Benson Muite)                                                                   |         |            |
| Reference contexts may now be named . . . . .                                                                | 3.7.10  |            |
| Added notfield step in Sourcemaps . . . . .                                                                  | ??      |            |
| <b>3.4 2016-05-10</b>                                                                                        |         |            |
| Added \ifcrossrefsource and \ifxrefsource . . . . .                                                          | ??      |            |
| Added data annotation feature . . . . .                                                                      | 3.6     |            |
| Added package option minxrefs . . . . .                                                                      | 3.1.2.1 |            |
| Added \ifuniqueprimaryauthor and associated global option . . .                                              | ??      |            |
| Added \DeprecateField, \DeprecateList and \DeprecateName ??                                                  |         |            |
| Added \ifcaselang . . . . .                                                                                  | ??      |            |
| Added \DeclareSortTranslit . . . . .                                                                         | ??      |            |
| Added uniquetitle test . . . . .                                                                             | ??      |            |
| Added \namelabeldelim . . . . .                                                                              | 3.10.1  |            |
| New starred variants of the \assignrefcontext* macros . . . . .                                              | 3.7.10  |            |
| New context-sensitive delimiter interface . . . . .                                                          | 3.10.2  |            |
| Moved prefixnumbers option to \newrefcontext and renamed to labelprefix . . . . .                            | 3.7.10  |            |
| Added \DeclareDatafieldSet . . . . .                                                                         | ??      |            |
| <b>3.3 2016-03-01</b>                                                                                        |         |            |
| Schema documentation for BibLaTeXML . . . . .                                                                | ??      | Biber only |
| Sourcemaping documentation and examples for BibLaTeXML . . . . .                                             | ??      | Biber only |
| Changes for name formats to generalise available name parts . . . . .                                        | ??      | Biber only |
| useprefix can now be specified per-namelist and per-name in BibLaTeXML datasources                           |         |            |
| New sourcemaping options for creating new entries dynamically and looping over map steps . . . . .           | ??      | Biber only |
| Added noalphaothers and enhanced name range selection in \DeclareLabelalphaTemplate . . . . .                | ??      |            |
| Added \DeclareDatamodelConstant . . . . .                                                                    | ??      | Biber only |
| Renamed firstinits and sortfirstinits                                                                        |         |            |
| Added \DeclareSortingNamekeyScheme . . . . .                                                                 | ??      |            |
| Removed messy experimental endnote and zoterordf support for Biber                                           |         |            |
| Added \nonameyeardelim . . . . .                                                                             | 3.10.1  |            |
| Added \extpostnotedelim . . . . .                                                                            | 3.10.1  |            |

### 3.2 2015-12-28

Added `pstrwidth` and `pcompound` to `\DeclareLabelalphaTemplate` ?? Biber only  
Added `\AtEachCitekey` . . . . . ??

### 3.1 2015-09

Added `\DeclareNolabel` . . . . . ?? Biber only  
Added `\DeclareNolabelwidthcount` . . . . . ?? Biber only

### 3.0 2015-04-20

Improved Danish (Jonas Nyrup) and Spanish (Iudenticus) translations  
`labelname` and `labeltitle` are now resolved by BibLaTeX instead of Biber for more flexibility and future extensibility  
New `\entryclone` sourcemap verb for cloning entries during sourcemaping ??  
New `\pernottype` negated per-type sourcemap verb . . . . . ??  
New range calculation command `\frangelen` . . . . . ??  
New bibliography context functionality . . . . . 3.7.10  
Name lists in the data model now automatically create internals for `\ifuse<name>` tests and booleans . . . . . 3.1.3.1 and . . . . . ??

### 2.9a 2014-06-25

`resetnumbers` now allows passing a number to reset to . . . . . 3.7.2

### 2.9 2014-02-25

Generalised shorthands facility . . . . . 3.7.3 Biber only  
Sorting locales can now be defined as part of a sorting scheme . . . . . ?? Biber only  
Added `sortinithash` . . . . . ?? Biber only  
Added Slovene localisation (Tea Tušar and Bogdan Filipič)  
Added `\mkbibitalic` . . . . . ??  
Recommend `begentry` and `finentry` bibliography macros . . . . . ??

### 2.8a 2013-11-25

Split option `language=auto` into `language=autocite` and `language=autobib` . . . . . 3.1.2.1 Biber only

### 2.8 2013-10-21

New `langidopts` . . . . . 2.2.3 Biber only  
`hyphenation` field renamed to `langid` . . . . . 2.2.3  
`polyglossia` support  
Renamed `babel` option to `autolang` . . . . . 3.1.2.1  
Corrected Dutch localisation  
Added `datelabel=year` option . . . . . 3.1.2.1  
Added `datelabelsource` field . . . . . ??

## 2.7a 2013-07-14

Bugfix - respect maxnames and uniquelist in `\finalandsemicolon`

Corrected French localisation

## 2.7 2013-07-07

Added field `eventtitleaddon` to default `datamodel` and styles . . . . . 2.2.2

Added `\ifentryinbib`, `\iffirstcitekey` and `\iflastcitekey` ??

Added `postpunct` special field, documented `multiprenote` and `multi`  
`postnote` special fields . . . . . ??

Added `\UseBibitemHook`, `\AtEveryMultiCite`, `\AtNextMultiCite`,  
`\UseEveryCiteHook`, `\UseEveryCitekeyHook`,  
`\UseEveryMultiCiteHook`, `\UseNextCiteHook`,  
`\UseNextCitekeyHook`, `\UseNextMultiCiteHook`,  
`\DeferNextCitekeyHook` . . . . . ??

Fixed `\textcite` and related commands in the numeric and verbose styles  
3.8.2

Added `multicite` variants of `\volcite` and related commands . . . . . 3.8.6

Added `\finalandsemicolon` . . . . . 3.10.3

Added citation delimiter `\textcitedelim` for `\textcite` and related  
commands to styles . . . . . ??

Updated Russian localization (Oleg Domanov)

Fixed Brazilian and Finnish localization

## 2.6 2013-04-30

Added `\printunit` . . . . . ??

Added field `clonesourcekey` . . . . . ?? Biber only

New options for `\DeclareLabelalphaTemplate` . . . . . ?? Biber only

Added `\DeclareLabeldate` and retired `\DeclareLabelyear` . . ?? Biber only

Added `nodate` localization string . . . . . ??

Added `\rangelen` . . . . . ??

Added starred variants of `\citeauthor` and `\Citeauthor` . . . . . 3.8.5

Restored original `url` format. Added `urlfrom` localization key . . . . . ??

Added `\AtNextBibliography` . . . . . ??

Fixed related entry processing to allow nested and cyclic related entries

Added Croatian localization (Ivo Pletikosić)

Added Polish localization (Anastasia Kandulina, Yuriy Chernyshov)

Fixed Catalan localization

Added smart “of” for titles to Catalan and French localization

Misc bug fixes

## 2.5 2013-01-10

Made `url` work as a localization string, defaulting to previously hard-coded value `,URL'`.

Changed some Biber option names to cohere with Biber 1.5.

New sourcemap step for conditionally removing entire entries . . . . . ?? Biber only

Updated Catalan localization (Sebastià Vila-Marta)

## 2.4 2012-11-28

Added `relatedoptions` field . . . . . ?? Biber only

Added `\DeclareStyleSourcemap` . . . . . ?? Biber only

Renamed `\DeclareDefaultSourcemap` to `\DeclareDriverSourcemap` ?? Biber only

Documented `\DeclareFieldInputHandler`,  
`\DeclareListInputHandler` and `\DeclareNameInputHandler`.

Added Czech localization (Michal Hoftich)

Updated Catalan localization (Sebastià Vila-Marta)

## 2.3 2012-11-01

Better detection of situations which require a Biber or  $\LaTeX$  re-run

New append mode for `\DeclareSourcemap` so that fields can be combined ?? Biber only

Extended auxiliary indexing macros

Added support for plural localization strings with `relatedtype` . . . . . ?? Biber only

Added `\csfield` and `\usefield` . . . . . ??

Added starred variant of `\usebibmacro` . . . . . ??

Added `\ifbibmacroundef`, `\iffieldformatundef`,  
`\iflistformatundef` and `\ifnameformatundef` . . . . . ??

Added Catalan localization (Sebastià Vila-Marta)

Misc bug fixes

## 2.2 2012-08-17

Misc bug fixes

Added `\revsdnamepunct` . . . . . 3.10.1

Added `\ifterseinits` . . . . . ??

## 2.1 2012-08-01

Misc bug fixes

Updated Norwegian localization (Håkon Malmedal)

Increased data model auto-loading possibilities . . . . . ?? Biber only

## 2.0 2012-07-01

Misc bug fixes

Generalised `singletitle` test a little . . . . . ?? Biber only

|                                                                                                         |         |            |
|---------------------------------------------------------------------------------------------------------|---------|------------|
| Added new special field <code>extratitleyear</code> . . . . .                                           | ??      | Biber only |
| Customisable data model . . . . .                                                                       | ??      | Biber only |
| Added <code>\DeclareDefaultSourceMap</code> . . . . .                                                   | ??      | Biber only |
| Added <code>labeltitle</code> option . . . . .                                                          | 3.1.2.3 | Biber only |
| Added new special field <code>extratitle</code> . . . . .                                               | ??      | Biber only |
| Made special field <code>labeltitle</code> customisable . . . . .                                       | ??      | Biber only |
| Removed field <code>reprinttitle</code> . . . . .                                                       | 3.4     | Biber only |
| Added related entry feature . . . . .                                                                   | 3.4     | Biber only |
| Added <code>\DeclareNoInit</code> . . . . .                                                             | ??      | Biber only |
| Added <code>\DeclareNoSort</code> . . . . .                                                             | ??      | Biber only |
| Added sorting option for <code>\printbibliography</code> and<br><code>\printshorthands</code> . . . . . | 3.7.2   | Biber only |
| Added <code>ids</code> field for citekey aliasing . . . . .                                             | 2.2     | Biber only |
| Added <code>sortfirstinits</code> option . . . . .                                                      | 3.1.2.3 | Biber only |
| Added data stream modification feature . . . . .                                                        | ??      | Biber only |
| Added customisable labels feature . . . . .                                                             | ??      | Biber only |
| Added <code>\citeyear*</code> and <code>\citedate*</code> . . . . .                                     | 3.8.5   |            |