

DANTE  
Deutschsprachige  
Anwendervereinigung T<sub>E</sub>X e.V.

Dominik Waßenhoven: *Bibliographien erstellen mit biblatex (Teil 2)*, Die T<sub>E</sub>Xnische Komödie 4/2008, S. 31–51.

Reproduktion oder Nutzung dieses Beitrags durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nichtkommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden. Für kommerzielle Nutzung ist die Zustimmung der Autoren einzuholen.

Die T<sub>E</sub>Xnische Komödie ist die Mitgliedszeitschrift von DANTE, Deutschsprachige Anwendervereinigung T<sub>E</sub>X e.V. Einzelne Hefte können von Mitgliedern bei der Geschäftsstelle von DANTE, Deutschsprachige Anwendervereinigung T<sub>E</sub>X e.V. erworben werden. Mitglieder erhalten Die T<sub>E</sub>Xnische Komödie im Rahmen ihrer Mitgliedschaft.

## Bibliographien erstellen mit biblatex (Teil 2)

Dominik Waßenhoven

Der Artikel stellt das Paket `biblatex` vor und ist aufgrund seines Umfangs zweigeteilt. Im ersten Teil, der in der *T<sub>E</sub>Xnischen Komödie* 2/2008 erschienen ist [1], wurde die allgemeine Benutzung von `biblatex` beschrieben. Im vorliegenden zweiten Teil wird anhand eines konkreten Beispiels aus dem geisteswissenschaftlichen Bereich gezeigt, wie man eigene Zitier- und Bibliographiestile erstellen kann. Da das Paket `biblatex` eine Fülle von Möglichkeiten bietet, kann und will dieser Artikel nicht erschöpfend sein; es sei deshalb explizit auf die sehr ausführliche Paketdokumentation [3] (in englischer Sprache) hingewiesen.

Der Artikel basiert auf der Version 0.7 von `biblatex`, mittlerweile ist bereits die Version 0.8 erschienen. Weitere Hinweise dazu finden sich am Ende dieses Beitrags.<sup>1</sup>

---

<sup>1</sup> Danken möchte ich Philipp Lehman, Simon Spiegel und Bernhard Tempel, die wiederum einige Fehler gefunden sowie viele Anregungen und Verbesserungsvorschläge gemacht haben. Alles, was jetzt noch falsch oder missverständlich sein sollte, habe ich selbst zu verantworten.

## Einleitung

Mit `biblatex` lassen sich recht einfach eigene Stile für die Literaturverweise im Text und für die Bibliographie erstellen. In diesem Artikel soll das grundsätzliche Vorgehen anhand eines konkreten Beispiels veranschaulicht werden, nämlich anhand der Vorgaben für die »Historische Zeitschrift« (HZ).<sup>2</sup> Es sei allerdings darauf hingewiesen, dass die Redaktion der HZ Manuskripte in Form von `TeX`-Dateien nicht selbst verarbeiten kann, sondern erst von der Setzerei in ein Word-kompatibles Format konvertieren lassen muss. Nichtsdestotrotz lohnt es sich, die Vorgaben der HZ nachzubauen, da sie in den Geschichtswissenschaften eine weite Verbreitung haben. Außerdem kann `biblatex` zusammen mit `tex4ht` [2] eingesetzt werden, um ein Word-kompatibles Format zu erzeugen.

Ein Großteil des Codes wird im Laufe dieses Artikels zu sehen sein, allerdings wäre es zu lang und zu uninteressant, den gesamten Code hier abzdrukken. Stattdessen wird er unter <http://biblatex.dominik-wassenhoven.de/dtk/> komplett angeboten. Die Ergebnisse dieses Artikels finden sich außerdem im Paket `historische-zeitschrift`, das vom CTAN heruntergeladen werden kann.<sup>3</sup>

## Was man beim Erstellen eigener Stile grundsätzlich beachten sollte

Ein Zitationsstil für `biblatex` besteht immer aus zwei Dateien, von denen eine für die Bibliographie zuständig ist (Endung `bbx`) und die andere für das Aussehen der Literaturangaben im Text (Endung `cbx`). Dazu kommen dann noch die eigentliche Paketdatei von `biblatex` (`biblatex.sty`), die uns hier weniger interessiert, und allgemeine Definitionen (`biblatex.def`), die auch innerhalb der Stildateien umdefiniert werden können. Die Datei `biblatex.def` ist übrigens ausführlich kommentiert; bei Fragen rund um die Formatierung ist es also ratsam, sich diese Datei anzusehen.

Beim Erstellen eines neuen Stils gibt es grundsätzlich zwei sinnvolle Vorgehensweisen.<sup>4</sup> Die erste Möglichkeit ist, einen bestehenden Stil einzubinden. Dazu stellt `biblatex` die Befehle `\RequireBibliographyStyle` und `\RequireCitationStyle` zur Verfügung. Will man etwa einen Stil kre-

<sup>2</sup> Diese Vorgaben lassen sich online unter <http://www.oldenbourg.de/verlag/historische-zeitschrift/hz-richtlinien.htm> abrufen.

<sup>3</sup> CTAN: `/macros/latex/expt1/biblatex-contrib/historische-zeitschrift`.

<sup>4</sup> Philipp Lehman hat sie selbst kurz in einem Beitrag in der Newsgroup »comp.text.tex« beschrieben (Message-ID: <ermsi1\$ab6\$1@online.de>).

ieren, dessen Verweise im Text genauso aussehen wie beim biblatex-Stil »authoryear«, erstellt man eine cbx-Datei, beispielsweise mit dem Namen »meinstil.cbx«, die folgenden Code enthält:

```
\ProvidesFile{meinstil.cbx}[2008/06/15 v1.0 Kurzbeschreibung]
\RequireCitationStyle{authoryear}
\endinput
```

Damit werden alle Informationen aus `authoryear.cbx` übernommen, man muss lediglich eigene Anpassungen ergänzen oder Befehle umdefinieren.

Je mehr Anpassungen nötig sind, desto problematischer ist diese Vorgehensweise allerdings, denn alle Änderungen, die an `authoryear.cbx` vorgenommen werden – etwa bei Updates –, betreffen dann auch `meinstil.cbx`. Deshalb ist die zweite Variante, die auf den ersten Blick etwas aufwändiger zu sein scheint, immer dann vorzuziehen, wenn viele Anpassungen nötig sind. Dazu erstellt man eine echte Kopie der Datei, von der man ausgehen möchte – beispielsweise `authoryear.bbx`. Der Kopie gibt man den Namen seines Stils, etwa `meinstil.bbx`, und ändert die Zeile mit `\ProvidesFile` entsprechend ab. Die Zeile `\RequireBibliographyStyle{standard}` wird gelöscht und stattdessen der Inhalt von `standard.bbx` eingefügt, wobei die `\ProvidesFile`-Zeile und der Befehl `\endinput` weggelassen werden. Die Datei `meinstil.bbx` ist damit eine vollständige und unabhängige Kopie von `authoryear.bbx`.

In beiden Fällen ist es zunächst sinnvoll, einen der vorhandenen Stile als Ausgangsbasis zu wählen. Der erste Schritt sollte also sein, einen Stil zu suchen, der dem gewünschten Ergebnis möglichst nahe kommt.

## Anpassungen für die »Historische Zeitschrift« (HZ)

### Auswahl des Ausgangsstils

Wie den bereits genannten »Richtlinien für die Gestaltung von Text und Anmerkungen in der Historischen Zeitschrift« (siehe Fußnote 2 auf Seite 2) zu entnehmen ist, folgt die HZ einem Autor-Titel-Schema. Da es sich um eine Zeitschrift handelt, fällt das eigentliche Literaturverzeichnis weg. Stattdessen wird beim ersten Literaturverweis ein Vollzitat gewünscht, in den folgenden Verweisen dann eine kurze Zitierung in der Form »Autorennachname, Titelstichwort (wie Anm. N, S)«, wobei »S« für die Seitenzahl steht. Die-

sen Anforderungen am nächsten kommt der von biblatex mitgelieferte Stil »verbose-inote«.<sup>5</sup>

Zunächst erstellt man also eine Kopie des Stils »verbose-inote«, wie ich es oben auf Seite 3 beschrieben habe, und nennt sie »historische-zeitschrift«. Bei `verbose-inote.bbx` fällt allerdings auf, dass die Datei lediglich den Bibliographiestil »authortitle« lädt. Man muss also `authortitle.bbx` kopieren und umbenennen. Außerdem sollte man nicht vergessen, den Inhalt von `standard.bbx` ebenfalls in die neue Datei `historische-zeitschrift.bbx` einzufügen.

So gerüstet kann man sich daran machen, die konkreten Änderungen in der Formatierung vorzunehmen. Dabei ist es sinnvoll, zwischen Anweisungen für die Zitate und solchen für das Literaturverzeichnis zu unterscheiden und sie in die jeweilige Datei zu schreiben (`cbx` für Zitate, `bbx` für die Bibliographie). Das ist allerdings nicht immer sauber zu trennen, weil manche Anweisungen für beide Bereiche gelten.

### Anpassung der Formatierung

Anders als in der Standardeinstellung von biblatex sehen die Richtlinien der HZ vor, dass der Titel nicht kursiv gesetzt wird. biblatex stellt für die Formatierung der meisten Felder einen speziellen Befehl bereit: `\DeclareFieldFormat`. Für die Titel in den Zitaten ist das Feldformat »citetitle« verantwortlich. In `biblatex.def`<sup>6</sup> sind diese Feldformate folgendermaßen definiert:

```
\DeclareFieldFormat{citetitle}{\mkbibemph{#1\isdot}}
\DeclareFieldFormat[article]{citetitle}{\mkbibquote{#1}}
```

Titel in Zitaten werden also kursiv gesetzt, Titel von Literaturangaben des Typs `@article` werden dagegen in Anführungszeichen gesetzt.<sup>7</sup> Diese Befehle ändern wir ab, so dass die Kursivierung oder das Einschließen in Anführungszeichen entfallen, und schreiben sie in `historische-zeitschrift.cbx`:

<sup>5</sup> In den Richtlinien ist nicht explizit davon die Rede, dass direkte Wiederholungszitate durch »ebd.« wiedergegeben werden. Die Praxis zeigt jedoch, dass das in der HZ so gehandhabt wird.

<sup>6</sup> Die meisten Formatierungen sind in der Datei `biblatex.def` festgelegt. Sofern ich im Folgenden keine anderen Angaben mache, ist davon auszugehen, dass die Definitionen aus dieser Datei stammen.

<sup>7</sup> `\mkbibemph` hebt den Text wie `\emph` hervor, `\mkbibquote` bedient sich des Pakets `csquotes` [4], um Anführungszeichen zu setzen. Gleichzeitig kümmern sich diese Befehle um anschließende Zeichensetzung. Beide Befehle sind in `biblatex.sty` definiert.

```
\DeclareFieldFormat{citetitle}{#1\isdot}
\DeclareFieldFormat[article]{citetitle}{#1}
```

Analog zu `@article` müssen auch die »citetitle« der Typen `@inbook`, `@incollection`, `@inproceedings`, `@patent`, `@thesis` und `@unpublished` angepasst werden.

Damit sind aber noch nicht alle Titelangaben richtig eingestellt, denn für die Vollzitate bedient sich biblatex der Definitionen des Literaturverzeichnisses. Deshalb müssen auch die Formatierungen für das Literaturverzeichnis geändert werden, diesmal in `historische-zeitschrift.bbx`:

```
\DeclareFieldFormat{title}{#1\isdot}
\DeclareFieldFormat[article]{title}{#1}
...
```

Damit sind alle Vorkommen des Feldes »title« abgedeckt. Es fehlen allerdings noch die Feldformate für Buchtitel bei `@incollection`-Einträgen, für Zeitschriftentitel und für Haupttitel bei mehrbändigen Werken, die ebenfalls in der `bbx`-Datei ergänzt werden sollten:

```
\DeclareFieldFormat{booktitle}{#1\isdot}
\DeclareFieldFormat{journaltitle}{#1\isdot}
\DeclareFieldFormat{issuetitle}{#1\isdot}
\DeclareFieldFormat{maintitle}{#1\isdot}
```

Die Richtlinien der HZ schreiben vor, dass Seitenzahlen ohne einleitendes »S. « erscheinen sollen. Dazu müssen die Feldformate für »postnote« und »multipostnote«<sup>8</sup> geändert werden:

```
\DeclareFieldFormat{postnote}{#1}
\DeclareFieldFormat{multipostnote}{#1}
```

Außerdem soll nach Namen ein Komma erscheinen, kein Punkt. Dafür ist der Befehl `\labelnamepunct` zuständig:

```
\renewcommand*\labelnamepunct{\addcomma\space}
```

Analog dazu realisieren wir die geforderten Schrägstriche zwischen mehreren Autoren oder Herausgebern und zwischen mehreren Orten:

```
\renewcommand*\multinamedelim{\addslash}
\renewcommand*\finalnamedelim{\addslash}
\renewcommand*\multilistdelim{\addslash}
```

<sup>8</sup> Die »multipostnote« kommt bei mehrfachen Zitationen mit einem Befehl (`\cites` etc.) zum Einsatz.

```
\renewcommand*{\finallistdelim}{\addslash}
```

`\final...delim` ist dabei jeweils der Trenner zwischen dem vorletzten und dem letzten Datensatz der Aufzählung. Diese Unterscheidung ist wichtig, wenn etwa Autoren in einer kommaseparierten Liste ausgegeben werden sollen, der letzte Name aber mit einem »und« angeschlossen wird. Außerdem gibt es noch den Befehl `\revsdnamedelim` (für *reversed name delimiter*), der immer dann zum Einsatz kommt, wenn etwa der erste Name als »Nachname, Vorname« ausgegeben wird, alle weiteren aber als »Vorname Nachname«. Es handelt sich dabei um einen Begrenzer, der *zusätzlich* zu `\multinamedelim` bzw. `\finalnamedelim` gesetzt wird. Da in der HZ Schrägstriche zwischen Namen gesetzt werden, kann `\revsdnamedelim` auf seinem Standard (leer) bleiben.

Die Autorennamen sollen in der HZ kursiv erscheinen. Anders als bei den übrigen Feldern eines BIB<sub>T</sub>E<sub>X</sub>-Eintrags wird die Formatierung der Namen nicht durch `\DeclareFieldFormat` eingestellt, sondern mit einigen Befehlen, in die für die Kursivierung `\textit` eingefügt werden muss. Die Standard-Definitionen finden sich abermals in der Datei `biblatex.def`. Da sich diese Definitionen sowohl auf das Literaturverzeichnis als auch auf die Zitate im Text beziehen, ist es in diesem Fall letztlich irrelevant, ob man sie in die `bbx`- oder die `cbx`-Datei einfügt.

```
\renewcommand*{\mkbibnamefirst}[1]{\textit{#1}}
\renewcommand*{\mkbibnamelast}[1]{\textit{#1}}
\renewcommand*{\mkbibnameprefix}[1]{\textit{#1}}
\renewcommand*{\mkbibnameaffix}[1]{\textit{#1}}
```

Das *prefix* ist der so genannte von-Teil des Namens, das *affix* ein nachgestellter Teil, beispielsweise der in amerikanischen Namen vorkommende »Jr.«.

### *Bibliography driver* und *bibmacros*

Bei der »Historischen Zeitschrift« ist es nicht vorgesehen, dass der Verlag ausgegeben wird. Das ist natürlich keine einfache Anpassung wie die Änderung des Schriftstils. Zunächst sollte also geklärt werden, wie `biblatex` eigentlich die Literaturangaben aus den einzelnen Feldern eines BIB<sub>T</sub>E<sub>X</sub>-Eintrags zusammensetzt. Dafür gibt es die *bibliography driver*, die mit dem Befehl `\DeclareBibliographyDriver` definiert werden. Alle von `biblatex` bereitgestellten *bibliography driver* finden sich in der Datei `standard.bbx`. Als Beispiel schauen wir uns einen Ausschnitt aus dem *bibliography driver* des Typs `@book` an:

```

\DeclareBibliographyDriver{book}{%
  \usebibmacro{bibindex}%
  \usebibmacro{author/editor}%
  \setunit{\labelnamepunct}\newblock
  ...
  \printlist{language}%
  \newunit\newblock
  \usebibmacro{byeditor+others}%
  \newunit\newblock
  \printfield{edition}%
  \newunit
  \iffieldundef{maintitle}
    {\printfield{volume}%
     \printfield{part}}
    {}%
  ...
  \newunit\newblock
  \usebibmacro{publisher+location+year}%
  ...

```

Zunächst wird mit dem Befehl `\usebibmacro` ein *bibmacro* aufgerufen, das an einer anderen Stelle definiert ist. Die Befehle `\newunit` und `\setunit` sorgen für die Zeichensetzung. `\newunit` stellt das Zeichen bereit, das mit `\newunitpunct` definiert wurde, in der Standardeinstellung ein Punkt. Mit `\setunit` kann man eigene Zeichen verwenden – dafür stellt biblatex Befehle zur Verfügung, beispielsweise `\adddot` (Abkürzungspunkt), `\addperiod` (Satzendepunkt), `\addcomma` (Komma), `\addcolon` (Doppelpunkt) oder `\addspace` (ein Leerschritt).<sup>9</sup> Man kann auch auf vorher definierte Makros zurückgreifen, so wie im Beispiel oben nach dem *bibmacro* »author/editor« `\labelnamepunct` aufgerufen wird. Bei der Verwendung von `\setunit` sollte man darauf achten, die Zeile mit einem Kommentarzeichen (%) zu beenden, damit keine überflüssigen Leerzeichen entstehen.<sup>10</sup> Daneben kann man in einem *bibliography driver* noch die Befehle `\printfield` und `\printlist` verwenden, die das entsprechende Feld ausgeben, allerdings nur, wenn sein Inhalt nicht leer ist.<sup>11</sup> Theoretisch wäre auch der Befehl `\printnames` möglich;

<sup>9</sup> Die Befehle finden sich im Abschnitt »Punctuation and spacing« der biblatex-Dokumentation [3]. Weitere Informationen zur Funktionsweise von `\newunit` und `\setunit` finden sich auch unten auf Seite 14.

<sup>10</sup> Im obigen Beispiel ist das lediglich deshalb nicht nötig, weil noch ein `\newblock` folgt. Näheres zu diesem Befehl und seiner Bedeutung ist in der biblatex-Dokumentation [3] im Abschnitt »Block and unit punctuation« erklärt.

<sup>11</sup> Listenfelder sind solche, die mehrere mit dem Kennwort »and« voneinander getrennte Elemente enthalten können, beispielsweise »location« oder »publisher«.

wegen der Komplexität der Namensausgaben findet sich dieser Befehl aber meist in eigenen *bibmacros* wieder, so wie im Beispiel bei »author/editor«. Schließlich können noch einige Bedingstests eingebunden werden, so etwa das `\iffieldundef{maintitle}`, das überprüft, ob das Feld »maintitle« in einem BibTeX-Eintrag gesetzt ist; im positiven Fall wird die erste Klammer, ansonsten die zweite Klammer ausgeführt.<sup>12</sup>

Der Verlag wird über das *bibmacro* »publisher+location+year« eingebunden, also sollten wir uns dieses *bibmacro* ansehen, das ebenfalls in `standard.bbx` definiert ist:

```
\newbibmacro*{publisher+location+year}{%
  \printlist{location}%
  \iflistundef{publisher}%      <--- entfernen
    {\setunit*{\addspace}}%    <--- entfernen
    {\setunit*{\addcolon\space}}% <--- entfernen
  \printlist{publisher}%      <--- entfernen
  \setunit*{\addcomma\space}%
  \printfield{year}%
  \newunit}
```

Man muss hier also nur die entsprechenden Zeilen, die den Herausgeber betreffen, entfernen. Im jeweiligen *bibliography driver* kann auch die Ausgabe ganzer Felder unterdrückt werden, ohne dass man sie aus der `bib`-Datei löschen muss. Für die Anpassungen an die HZ trifft das auf die Felder ISBN und DOI zu, die wir komplett aus jedem *bibliography driver* herausnehmen, indem wir folgende Zeilen löschen (oder auskommentieren):

```
\printfield{isbn}%
\newunit\newblock
\printfield{doi}%
\newunit\newblock
```

## Namensformate

Es gibt noch ein Problem mit den Namen von Herausgebern, denn bei unselbstständigen Schriften (also wenn etwa ein Beitrag in einem Sammelband zitiert wird) sieht die HZ keine Kursivierung vor. Diese Formatierung lässt sich nicht so leicht einstellen wie die Titel bei unterschiedlichen Literaturtypen – was vor allem daran liegt, dass die Namen von `biblatex` speziell verarbeitet werden müssen, weil es sich um Listen mit unterschiedlichen Bestandteilen

<sup>12</sup>Näheres zu den verfügbaren Tests findet sich in der `biblatex`-Dokumentation [3], Abschnitt »Stand-alone tests«.



handelt (Vorname, Nachname, von-Teil, Junior-Teil). Statt der Feldformate gibt es Namensformate, die allerdings etwas komplexer sind. Der Befehl `\DeclareNameFormat` kann insgesamt acht unterschiedliche Argumente enthalten, die für folgende Teile des Namens stehen:

#1 Nachname	#2 Nachname (Initialen)
#3 Vorname	#4 Vorname (Initialen)
#5 Präfix ( <i>der »von«-Teil</i> )	#6 Präfix (Initialen)
#7 Affix ( <i>der »Junior«-Teil</i> )	#8 Affix (Initialen)

Für die abweichende Formatierung der Herausgebernamen in unselbständigen Schriften müssen wir also ein eigenes Namensformat erstellen. Dazu schauen wir uns zunächst einmal an, wie biblatex das Namensformat in Zitaten gestaltet. In `biblatex.def` finden wir:

```
% The default used by \citename
\DeclareNameAlias{citename}{default}
```

Diese Zeile bedeutet nichts anderes, als dass das Namensformat »default« verwendet wird. Das wiederum ist folgendermaßen definiert:

```
\DeclareNameFormat{default}{%
  \usebibmacro{name: first-last}{#1}{#3}{#5}{#7}%
  \usebibmacro{name: andothers}}
```

Es werden also zwei *bibmacros* aufgerufen, wobei die Makronamen für sich sprechen. Zunächst werden die Namen in der Reihenfolge »Vorname Nachname« (`name: first-last`) aufgerufen, wobei als Argumente jeweils die vollständigen Namensbestandteile übergeben werden. Anschließend wird das Makro `name: andothers` aufgerufen, das nur zur Anwendung kommt, wenn die maximale Anzahl von Namen, die ausgegeben werden soll, überschritten wird. Der Wert, nach dem sich das Makro richtet, wird durch die Paketoption »maxnames« eingestellt.

Wir kopieren also diese Definition und ändern sie leicht ab:

```
\DeclareNameFormat{upshape}{%
  \usebibmacro{name: first-last-upshape}{#1}{#3}{#5}{#7}%
  \usebibmacro{name: andothers}}
```

Damit wird nun auf ein neues *bibmacro* verwiesen, das wir natürlich erst definieren müssen. Auch dazu gehen wir von dem vorhandenen Makro `name: first-last` aus:

```
\newbibmacro*{name: first-last}[4]{%
  \usebibmacro{name: delim}{#2#3#1}%
  \ifblank{#2}{\mkbibnamefirst{#2}\isdot\addlowpenspace}%
```

```

\ifblank{#3}{}{%
  \mkbibnameprefix{#3}\isdot
  \ifpunctmark{'}
  {}
  {\ifuseprefix{\addhighspace}{\addlowspace}}}%
\mkbibnamelast{#1}\isdot
\ifblank{#4}{}{\addlowspace\mkbibnameaffix{#4}\isdot}}

```

Die wichtigen Teile sind grau hinterlegt. Diese Definitionen hatten wir oben geändert, um die Namen kursiv darzustellen. Wir nehmen diese Befehle nun aus unserem neuen *bibmacro* heraus:

```

\newbibmacro*{name: first-last-upshape}[4]{%
  \usebibmacro{name:delim}{#2#3#1}%
  \ifblank{#2}{}{#2\isdot\addlowspace}%
  \ifblank{#3}{}{%
    #3\isdot
    \ifpunctmark{'}
    {}
    {\ifuseprefix{\addhighspace}{\addlowspace}}}%
  #1\isdot
  \ifblank{#4}{}{\addlowspace#4\isdot}}

```

Jetzt brauchen wir noch ein neues *bibmacro*, das wir bei den Typen `@incollection` und `@inproceedings` anstelle des *bibmacros* »editor« in den jeweiligen *bibliography driver* einfügen können. Es ist eine Kopie des *bibmacros* »editor«, in der dem Befehl `\printnames` das vorhin definierte Namensformat »upshape« als Option mitgegeben wird:

```

\newbibmacro*{editorincoll}{%
  \ifthenelse{\ifuseeditor\AND\NOT\ifnameundef{editor}}
    {\ifthenelse{\iffieldequals{author}{editor}\AND
      \NOT\iffirstonpage}
      {\bibnamedash}
      {\printnames [upshape]{editor}\addspace
        \savefield{namehash}{\bbx@lasthash}}}%
  \usebibmacro{editorstrg}%
  \clearname{editor}}
{\global\undef\bbx@lasthash}}

```

Jetzt müssen wir noch den Befehl `\usebibmacro{editorincoll}` beim *bibliography driver* für `@incollection` und `@inproceedings` einfügen. Da die Herausgeber nicht, wie es bei biblatex Standard ist, nach dem Buchtitel erscheinen sollen, sondern bereits davor, löschen wir die beiden folgenden Zeilen

```

\newunit\newblock
\usebibmacro{byeditor+others}%

```

und fügen etwas weiter oben stattdessen zwei neue Zeilen ein:

```
\usebibmacro{in:}%
\usebibmacro{editorincoll}% <--- NEU
\setunit{\labelnamepunct}\newblock% <--- NEU
\usebibmacro{maintitle+booktitle}%
```

### Ändern feststehender Ausdrücke (*Bibliography strings*)

Jetzt sind die Herausgeber fast fertig. Eine Kleinigkeit ist allerdings noch nicht so, wie von der HZ gefordert. Denn nach dem Namen des Herausgebers erscheint derzeit noch ein », Hg.«. Dort soll aber »(Hrsg.)« erscheinen, also der Begriff in Klammern, dafür ohne Komma davor. Feststehende Begriffe, so genannte *bibliography strings*, lassen sich bei biblatex mit dem Befehl `\DeclareBibliographyStrings` definieren. Man muss nur wissen, welcher Begriff wofür steht. Am einfachsten schaut man dazu in die `lbx`-Datei der Sprache, die man verwendet. In `german.lbx`<sup>13</sup> findet sich dann:

```
\DeclareBibliographyStrings{%
...
editor      = {{\typeeditor}{\typeeditor}},
editors     = {{\typeeditors}{\typeeditors}},
typeeditor  = {{Herausgeber}{Hg\addot}},
typeeditors = {{Herausgeber}{Hg\addot}},
...
}
```

Der Befehl `\DeclareBibliographyStrings` kann nur in `lbx`-Dateien angewendet werden. Er enthält eine Liste mit `key=value`-Definitionen, wobei der Wert zwei Varianten hat: Die erste Variante ist immer die ausgeschriebene, die zweite eine Abkürzung. Anders ist es hingegen bei dem Befehl `\DefineBibliographyStrings`, der in den `bbx`- und `cbx`-Dateien verwendet wird: Er nimmt als Argument eine Sprache<sup>14</sup> und dann eine Liste mit Einträgen des Schemas `key=value`, wobei immer nur ein Wert angegeben werden kann. Man muss sich hier also entscheiden, ob man eine abgekürzte oder eine ausgeschriebene Variante braucht. Da »editor«, wie man oben sieht, lediglich auf »typeeditor« verweist, sollte man Letzteres abändern:

```
\DefineBibliographyStrings{german}{%
typeeditor  = {Hrsg\addot},
```

<sup>13</sup> Die Datei `ngerman.lbx` für neue deutsche Rechtschreibung lädt `german.lbx` und definiert lediglich einige wenige Ausdrücke um.

<sup>14</sup> Eine Sprache muss `\DeclareBibliographyStrings` hingegen nicht mitgegeben werden, weil dieser Befehl ohnehin in einer sprachspezifischen Datei steht, eben den `lbx`-Dateien.

```
typeeditors = {Hrsg\adddot},
}
```

Jetzt haben wir zwar den Ausdruck in »Hrsg.« geändert, aber er steht noch nicht in Klammern. Dazu muss das *bibmacro* »editorstrg« abgeändert werden. Die Originaldefinition sieht folgendermaßen aus:

```
\newbibmacro*{editorstrg}{%
  \ifthenelse{\value{editor}>1\OR\ifandothers{editor}}
    {\bibstring{editors}}
    {\bibstring{editor}}}
```

Der Befehl `\bibstring` greift auf den vorher undefinierten *bibliography string* zu (genauer gesagt auf »editor«, das ja wiederum auf »typeeditor« verweist). Hier ist es am besten, sich des Befehls `\mkbibparens` zu bedienen, dessen Argument bei der Ausgabe in runde Klammern eingeschlossen wird:

```
\renewbibmacro*{editorstrg}{%
  \ifthenelse{\value{editor}>1\OR\ifandothers{editor}}
    {\mkbibparens{\bibstring{editors}}}
    {\mkbibparens{\bibstring{editor}}}
```

Es bleibt nur noch, das Komma vor dem Ausdruck »(Hrsg.)« zu entfernen. Dazu müssen wir das *bibmacro* »editor« anpassen. Nach `\printnames{editor}` steht dort `\addcomma\space`, das durch `\addspace` ersetzt werden muss.

Für den Fall, dass ein Buch einen Autor und einen Herausgeber hat, sollten ebenfalls Vorkehrungen getroffen werden, denn bei Werkausgaben wird der Herausgeber stets nachgestellt, wobei er durch »Hrsg. v.« eingeleitet wird.<sup>15</sup> Dazu passen wir die entsprechenden *bibliography strings* an:

```
bytypeeditor = {hrsg\adddot},
byeditor     = {\bytypeeditor\addabbrvspace v\adddot},
```

Der Befehl `\addabbrvspace` fügt dabei einen Leerschritt ein, an dem grundsätzlich umbrochen werden kann. Dies lässt sich mit dem Zähler `abbrvpenalty` steuern. Setzt man ihn auf »0«, kann an dieser Stelle immer umbrochen werden. Setzt man den Zähler auf »10000« oder höher, wird ein Umbruch in jedem Fall vermieden. Das kann allerdings zu *overflow boxes* führen, also zu Text, der über den rechten Textrand hinausragt. Das ist bei Literaturverzeichnissen problematisch, da man hier selten umformulieren

<sup>15</sup> Das geht aus den Richtlinien zwar nicht hervor, wurde mir auf Nachfrage bei der Redaktion aber so mitgeteilt.

kann, um die *overflow box* zu vermeiden. Deshalb ist der Standardwert von `abbrvpenalty` recht niedrig eingestellt.<sup>16</sup>

Um eine konsistente Darstellung zu erhalten, sollten auch noch alle anderen *bibliography strings*, die ein »von« enthalten, durch die abgekürzte Form »v.« ersetzt werden.

### Änderung des Reihentitels (oder: Noch ein *bibmacro*)

In ähnlicher Weise erfolgt auch die Anpassung der Reihenangabe, die bei der HZ in Klammern stehen soll, wobei die Reihennummer mit »Bd.« eingeleitet wird. Wir schauen uns also das *bibmacro* »series+number« an:

```
\newbibmacro*{series+number}{%
  \printfield{series}%
  \setunit*{\addspace}%
  \printfield{number}%
  \newunit}
```

Es werden lediglich Reihentitel und Nummer gesetzt, getrennt durch einen Leerschritt. Die Sternvariante von `\setunit` bedeutet übrigens, dass überprüft wird, ob der letzte Befehl etwas ausgegeben hat;<sup>17</sup> falls nicht, wird auch `\setunit*` nicht ausgeführt. Da der gesamte Ausdruck in Klammern erscheinen soll, muss zunächst geprüft werden, ob das Feld »series« im BIBTEX-Eintrag überhaupt vorhanden ist. Falls nicht, soll natürlich auch keine Klammer gesetzt werden. Anschließend benutzen wir den Befehl `\printtext`, dem wir als optionales Argument »parens« mitgeben, womit das gesamte obligatorische Argument von `\printtext` in Klammern eingefasst wird. Dann muss man noch eine Abfrage einbauen, ob das Feld »number« (die Reihennummer) definiert ist, und für den positiven Fall ein Komma und den Ausdruck »Bd.« (*bibliography string* »volume«) zwischen Reihentitel und Nummer einfügen. Lässt man diese Abfrage weg, steht », Bd. « auch da, falls es einen Reihentitel, aber keine Nummer gibt. Schließlich wird der gesamte Ausdruck mit einem Punkt beendet. Damit sieht das *bibmacro* folgendermaßen aus:

```
\newbibmacro*{series+number}{%
  \iffieldundef{series}%
  {}
```

<sup>16</sup> `biblatex` nutzt den Wert von `\hyphenpenalty`, der normalerweise »50« beträgt. Weitere Erläuterungen zu diesem Zähler sowie anderen, die `biblatex` zur Verfügung stellt, stehen in der Dokumentation [3] im Abschnitt »Lengths and counters«.

<sup>17</sup> Dabei werden folgende Befehle berücksichtigt: `\printtext`, `\printfield`, `\printlist`, `\printnames` und `\bibstring`.

```
{\printtext [parens]{%
\printfield{series}%
\iffieldundef{number}
  {}
  {\setunit{\addcomma\space}%
\bibstring{volume}~\printfield{number}}%
\addot}}
```

Dabei bleibt allerdings ein Problem bestehen, denn jetzt wird auch nach der schließenden Klammer ein Punkt ausgegeben. Der wird im *bibliography driver* durch das `\newunit\newblock` nach dem Befehl `\usebibmacro{series+number}` erzeugt. Eine einfache Ersetzung durch `\setunit*` (das ja nur aufgerufen wird, wenn der Befehl *vorher* eine Ausgabe erzeugt hat) reicht aber nicht aus. Am besten lässt sich das am L<sup>A</sup>T<sub>E</sub>X-Code selbst demonstrieren. Als Beispiel habe ich den *bibliography driver* für @book gewählt; bei den übrigen Typen kann es etwas anders aussehen. Der entsprechende Abschnitt sieht in der Originalversion folgendermaßen aus:

```
\usebibmacro{series+number}%
\newunit\newblock
\printfield{note}%
\newunit\newblock
\usebibmacro{publisher+location+year}%
```

Wenn man nur die Zeichensetzung nach »series+number« ändert,

```
\usebibmacro{series+number}%
\setunit*{\addspace}%
\printfield{note}%
\newunit\newblock
\usebibmacro{publisher+location+year}%
```

passiert Folgendes: Der Reihentitel wird gesetzt. Wenn das Feld »note« vorhanden ist, wird `\setunit*` ausgeführt, es wird also ein Leerschritt eingefügt. Anschließend geht es normal weiter – wunderbar. *Aber:* Wenn das Feld »note« nicht vorhanden ist, wird zunächst der Befehl `\newunit\newblock` abgearbeitet, bevor das *bibmacro* »publisher+location+year« ausgeführt wird,<sup>18</sup> und schon haben wir wieder einen Punkt hinter der Klammer mit dem Reihentitel – nicht gut. Wir müssen also mit mehreren `\setunit*`-Befehlen arbeiten, zumindest bis »publisher+location+year«, da man in der Regel davon ausgehen kann, dass Ort und Jahr in jeder Literaturangabe vorhanden sind:

<sup>18</sup> `\newunit` und `\setunit` (ohne Stern) werden erst dann ausgeführt, wenn der *nachfolgende* Befehl eine Ausgabe erzeugt. Mehrere `\newunit`-Befehle zwischen leeren Feldern erzeugen also nicht mehrere Punkte.

```

\usebibmacro{series+number}%
\setunit*{\addspace}%
\printfield{note}%
\setunit*{\newunitpunct}%
\usebibmacro{publisher+location+year}%

```

Und da wir gerade schon dabei sind, können wir an dieser Stelle noch die Auflage einfügen, die bei der HZ nämlich nach der Reihe erscheint, nicht davor:

```

\usebibmacro{series+number}%
\setunit*{\addspace}%
\printfield{edition}%
\setunit*{\newunitpunct}%
\printfield{note}%
\setunit*{\newunitpunct}%
\usebibmacro{publisher+location+year}%

```

### Anpassung der Zeitschriften

Bei Zeitschriften soll das »in:« vor dem Titel der Zeitschrift klein sein, davor soll ein Komma stehen. Das ist wiederum über den *bibliography driver* anzupassen, in diesem Fall für den Typ `@article`. Das `\newunit\newblock` vor dem *bibmacro* »in:« muss zu einem `\setunit{\addcomma\space}` geändert werden. Das Schöne ist, dass biblatex das »in:« jetzt automatisch klein setzt, weitere Anpassungen sind also nicht nötig.

Die Jahresangabe von Zeitschriften soll nicht in Klammern erfolgen, sondern umgeben von Kommata. Dazu muss der Befehl `\printtext[parens]` aus dem *bibmacro* »issue+date« entfernt werden. Die Kommata werden dann im *bibmacro* »journal+issuetitle« eingefügt:

```

... \setunit{\addcomma\space}%
    % statt: \setunit{\addspace}
\usebibmacro{issue+date}%
\setunit*{\addcomma\space}%
    % statt: \newunit\newblock ...

```

Man kann hier übrigens `\space` oder `\addspace` gleichermaßen verwenden. Der Unterschied liegt darin, dass die `\add...`-Befehle automatisch vorangehenden Leerraum mit Hilfe von `\unspace` löschen. Da das hier nicht nötig ist, können wir einfach `\space` verwenden. Analog ändern wir für `@inbook`, `@incollection` und `@inproceedings` die Zeichensetzung vor den Seitenzahlen von einem Punkt in ein Komma:

```
... \setunit*{\addcomma\space}%
    % statt: \newunit\newblock
    \usebibmacro{chapter+pages}% ...
```

Nun ist es fast geschafft. Bei Dissertationsschriften soll allerdings die Hochschule, an der die Arbeit verfasst wurde, nicht mit einem Komma von der Jahreszahl getrennt werden. Deshalb muss das *bibmacro* »institution+location+year« angepasst werden:

```
\newbibmacro*{institution+location+year}{%
  \printlist{location}%
  \iflistundef{institution}
    {\setunit*{\addspace}}
    {\setunit*{\addcolon\space}}%
  \printlist{institution}%
  \setunit*{\addspace}% <--- statt: \addcomma\space
  \printfield{year}%
  \newunit}
```

Damit nicht einfach »Diss.«, sondern »Diss. phil.« erscheint, muss übrigens in der *bib*-Datei das Feld »type« eingesetzt werden, in diesem Fall `type = {Diss. phil.}`. Damit ist auch gewährleistet, dass Dissertationen anderer Fakultäten, etwa eine medizinhistorische Abhandlung, korrekt zitiert werden können.

### Wiederholungszitate

Eine Sache muss noch angepasst werden, damit die Richtlinien der HZ vollständig umgesetzt sind. Die Wiederholungszitate haben derzeit noch die Form »Autor, s. Anm. N, S« (wobei mit »S« die Seitenzahl gemeint ist). Gefordert ist aber folgendes Schema: »Autor, Kurztitel (wie Anm. N, S)«. Da es sich hier ausschließlich um die Zitate im Text handelt, sollten die Änderungen in der *cbx*-Datei vorgenommen werden.

Als erstes passen wir den *bibliography string* an:

```
\DefineBibliographyStrings{german}{%
  seenote = {wie Anm\addot},
}
```

Nun müssen wir noch dafür sorgen, dass der Ausdruck von Klammern umgeben wird, wobei Seitenzahlen (oder auch andere Dinge, die im optionalen Argument eines `\cite`-Befehls übergeben werden) innerhalb des Klammerspaars ausgegeben werden sollen. Es reicht also nicht, den *bibliography string* »seenote« in Klammern zu setzen, denn dann würde die »postnote«, die das



optionale Argument eines `\cite`-Befehls ausgibt, erst nach der Klammer erscheinen. Es muss also ein Weg gefunden werden, innerhalb der »postnote« zu unterscheiden, ob ein Verweis auf eine andere Fußnote ausgegeben wird oder nicht (denn in Erstzitationen darf nach der »postnote« keine Klammer stehen). Um das überprüfen zu können, definieren wir uns einen neuen `if`-Schalter, der die Werte `true` oder `false` annehmen kann. Das Paket `etoolbox` [5], das von `biblatex` benötigt wird, bietet dafür den Befehl `\newbool` an:

```
\newbool{cbx@seenote}
```

Grundsätzlich sollte dieser Schalter auf »false« stehen. Das müssen wir `biblatex` mitteilen, am besten in den `bibmacros`, die das grundsätzliche Zitieren handhaben:

```
\newbibmacro*{cite}{%
  \global\boolfalse{cbx@seenote}% <--- NEU
  \iffieldundef{shorthand}
    {\usedriver
      {\DeclareNameAlias{sortname}{default}}
      {\thefield{entrytype}}}
    {\printfield{shorthand}}}
```

Exkurs: Die `\cite`-Befehle bei »verbose-inote«

Um das `bibmacro` »cite« besser verstehen zu können, folgt eine kurze Erläuterung, wie die `\cite`-Befehle im Stil »verbose-inote« aufgebaut sind. Dazu schauen wir uns zunächst die Definition des Zitierbefehls selbst an:

```
\DeclareCiteCommand{\cite}
  {\usebibmacro{prenote}}
  {\usebibmacro{citeindex}%
   \iffootnote
     {\usebibmacro{footcite}}
     {\usebibmacro{cite}}}
  {\multicitedelim}
  {\usebibmacro{cite:postnote}}
```

Ich will den Aufbau von `\DeclareCiteCommand` jetzt nicht in allen Einzelheiten durchkauen.<sup>19</sup> Wichtig ist in unserem Zusammenhang, dass mittels `\iffootnote` überprüft wird, ob man sich gerade in einer Fußnote befindet; falls ja, wird das `bibmacro` »footcite« ausgeführt, ansonsten »cite«. Das `bibmacro` »cite« führt dann ein Vollzitat aus, es sei denn, im `BIBTEX`-Eintrag

<sup>19</sup>Näheres dazu entnehmen Sie bitte der `biblatex`-Dokumentation [3], Abschnitt »Citation style files«.

wurde eine Sigle definiert, das Feld `shorthand` wurde mit Inhalt gefüllt. Der Grund für die Unterscheidung zwischen Text und Fußnoten ist, dass im Text auf keinen Fall ein normales Erstzitat erfolgen soll, denn ein späteres Zitat verweist immer auf die Fußnotennummer des Erstzitats – und wenn das im Text stünde, liefe die Referenz auf das automatisch erzeugte Label ins Leere. Da also grundsätzlich Zitate im Text möglich sind, diese dann aber keinen Verweis »wie Anm. N« aufweisen, muss die »postnote« in diesem Fall ohne Klammer gesetzt werden. Aus diesem Grund haben wir in das *bibmacro* »cite« die Zeile `\global\boolfalse{cbx@seenote}` eingefügt.

Damit sind wir wieder zurück beim if-Schalter, den wir auch in das *bibmacro* »footcite« einbauen müssen (für den Fall, dass es sich um ein Erstzitat handelt):

```
\newbibmacro*{footcite}{%
  \global\boolfalse{cbx@loccit}%
  \global\boolfalse{cbx@seenote}% <--- NEU
  \ifciteseen
    {\iffieldundef{shorthand}
      {\ifthenelse{\ifciteibid\AND\NOT\iffirstonpage}
        {\usebibmacro{footcite:ibid}}
        {\usebibmacro{footcite:note}}}
      {\usebibmacro{footcite:shorthand}}}
    {\usebibmacro{footcite:full}}%
  \usebibmacro{footcite:save}}
```

Das `\global` ist übrigens nötig, damit die Anweisung auch über das *bibmacro* selbst hinaus noch Gültigkeit besitzt. Das *bibmacro* »footcite« überprüft zunächst, ob der Eintrag bereits zitiert wurde (`\ifciteseen`). Falls ja (und keine Sigle definiert ist), wird überprüft, ob es sich um dasselbe Werk handelt wie das unmittelbar vorher zitierte (`\ifciteibid`); gleichzeitig wird sichergestellt, dass es nicht das erste Zitat auf der aktuellen Seite ist (`\AND\NOT\iffirstonpage`). Nur wenn diese Bedingungen erfüllt sind, wird das *bibmacro* »footcite:ibid« ausgeführt, das ein »ebd.« ausgibt. Ansonsten kommt das *bibmacro* »footcite:note« zum Einsatz. Wenn es sich dagegen um das Erstzitat handelt, wird auf das *bibmacro* »footcite:full« verwiesen.

Uns interessiert wegen des Fußnotenverweises das *bibmacro* »footcite:note«, das in der Standarddefinition folgendermaßen aussieht:

```
\newbibmacro*{footcite:note}{%
  \ifnameundef{labelname}
    {\printfield{label}}
    {\printnames{labelname}}%
  \ifsingletitle
```

```

{}
{\setunit*{\addcomma\space}%
 \printfield[title]{labeltitle}}%
\setunit*{\addcomma\space}%
\printtext{%
 \bibhyperlink{\thefield{entrykey}}:%
 \@nameuse{cbx@first@\thefield{entrykey}}}%
 \bibstring{seenote}\space
 \@nameuse{cbx@note@\thefield{entrykey}}}}

```

Ohne die Definitionen im Einzelnen zu besprechen (und damit den Rahmen dieses Artikels und die Kenntnisse seines Autors zu überschreiten), sorgen wir als Erstes dafür, dass der if-Schalter auf »true« gesetzt wird, da hier definitiv der Verweis auf eine frühere Fußnote ausgegeben werden muss:

```

\newbibmacro*{footcite:note}{%
 \global\booltrue{cbx@seenote}} ...

```

Vor der Ausgabe des *bibliography strings* »seenote« muss eine öffnende runde Klammer (`\bibleftparen`) eingefügt werden. Gleichzeitig soll aber kein Komma vor der Klammer stehen:

```

...\setunit*{\addspace}% <--- statt: \addcomma\space
\printtext{%
 \bibhyperlink{\thefield{entrykey}}:%
 \@nameuse{cbx@first@\thefield{entrykey}}}%
 \printtext{\bibleftparen}% <--- NEU
 \bibstring{seenote}\space ...

```

Eine schließende Klammer fügen wir hier nicht ein, denn möglicherweise folgt ja noch eine Seitenzahl, die ebenfalls vor der schließenden Klammer erscheinen soll. Hier kommt der if-Schalter ins Spiel, den wir jetzt im *bibmacro* »postnote« benutzen. Die Standarddefinition steht in `biblatex.def`:

```
\newbibmacro*{postnote}{%
  \iffieldundef{postnote}
  {}
  {\postnotedelim
   \printfield{postnote}}}
```

Wir kopieren das *bibmacro* in die *cbx*-Datei und definieren es um:

```
\renewbibmacro*{postnote}{%
  \ifbool{cbx@seenote}
  {\iffieldundef{postnote}
   {\printtext{\bibrightparen}}
   {\postnotedelim
    \printfield{postnote}%
    \printtext{\bibrightparen}}}}
  {\iffieldundef{postnote}
   {}
   {\postnotedelim
    \printfield{postnote}}}}
```

Nur wenn die `\ifbool`-Abfrage ein »true« zurückgibt (also jedesmal, wenn das *bibmacro* »footcite:note« aufgerufen wird, aber auch nur dann), wird die erste Klammer ausgeführt. Dort fügen wir eine schließende runde Klammer ein (`\bibrightparen`), die entweder direkt gesetzt wird oder nach der »postnote«, falls eine vorhanden ist. Wenn die `\ifbool`-Abfrage zu einem »false« führt (bei Erstzitat und bei Zitaten außerhalb von Fußnoten), soll dagegen keine schließende Klammer gesetzt werden. Diesen Teil belassen wir so wie vorher das gesamte *bibmacro* »postnote« war.

Jetzt fehlt nur noch der Kurztitel, der in den Wiederholungszitaten nach dem Autor erscheinen soll. Hierzu können wir eine Option benutzen, die *biblatex* bietet: »singletitle«. Optionen lassen sich auch in *bbx*- und *cbx*-Dateien einstellen, nicht nur beim Laden des Pakets. In unserer *cbx*-Datei werden auch schon einige Optionen aufgerufen:

```
\ExecuteBibliographyOptions{
  citetracker=context,
  ibidtracker=constrict,
  singletitle,
  pagetracker,
  autocite=footnote
}
```

Wie man sieht, ist die Option »singletitle« auf »true« gesetzt (`singletitle` und `singletitle=true` sind bedeutungsgleich). Das heißt, dass *biblatex* den Kurztitel nur dann ausgibt, wenn von einem Autor mehr als ein Werk zitiert

wurde. Die Option muss also auf »false« gesetzt werden, damit diese Unterscheidung nicht getroffen wird und der Kurztitel in Folgezitatzen auch dann ausgegeben wird, wenn ein Autor nur mit einem Werk vertreten ist – so wie es die Richtlinien der HZ fordern.

```
\ExecuteBibliographyOptions{
...
singletitle=false,
...
}
```

Damit sind die Anpassungen für die »Historische Zeitschrift« fertig.

## Zusammenfassung

Wie hoffentlich gezeigt werden konnte, lassen sich Literaturverweise und Bibliographien mit biblatex sehr flexibel anpassen. Dabei reichen solide L<sup>A</sup>T<sub>E</sub>X-Kenntnisse und ein wenig Beschäftigung mit dem Aufbau und der Funktionsweise von biblatex aus. Ich will noch einmal darauf hinweisen, dass mit dem gezeigten Beispiel nur ein kleiner Teil der Möglichkeiten, die biblatex bietet, gezeigt werden konnte. Ich hoffe aber, dass die grundsätzliche Herangehensweise beim Arbeiten mit biblatex verständlich geworden ist.

*Ein Hinweis zum Schluss: Wie bereits eingangs erwähnt, basiert dieser Artikel auf der Version 0.7 von biblatex, die mittlerweile bereits überholt ist. Das grundsätzliche Vorgehen, um das es mir vor allem ging, bleibt aber bestehen. Der zugrunde liegende Stil »verbose-inote« hat sich im Zuge der Aktualisierung geändert; die gezeigten Ausschnitte können also abweichen. Ich musste allerdings nur eine Änderung vornehmen, um den hier entwickelten Stil auch unter biblatex 0.8 einsetzen zu können: Das bibmacro »finentry«, das bei biblatex 0.7 noch in biblatex.def stand, findet sich jetzt in standard.bbx und musste deshalb in historische-zeitschrift.bbx nachgetragen werden.*

## Literatur

- [1] Dominik Waßenhoven: *Bibliographien erstellen mit biblatex (Teil 1); Die T<sub>E</sub>Xnische Komödie*; 2, S. 53–75; Mai 2008.
- [2] Eitan M. Gurari: *T<sub>E</sub>X4ht: L<sup>A</sup>T<sub>E</sub>X and T<sub>E</sub>X for Hypertext*; Okt. 2008; CTAN: [support/TeX4ht](http://www.cse.ohio-state.edu/~gurari/TeX4ht), URL: <http://www.cse.ohio-state.edu/~gurari/TeX4ht>.
- [3] Philipp Lehman: *The biblatex package. Programmable bibliographies and citations*; Okt. 2008; CTAN: [macros/latex/exptl/biblatex](http://www.ctan.org/tex-archive/macros/latex/exptl/biblatex).

- [4] Philipp Lehman: *The csquotes package. Context sensitive quotation facilities*; Okt. 2008; CTAN: [macros/latex/contrib/csquotes](#).
- [5] Philipp Lehman: *The etoolbox package. An  $\epsilon$ -T<sub>E</sub>X toolbox for class and package authors*; Okt. 2008; CTAN: [macros/latex/contrib/etoolbox](#).