

# The biblatex-chicago-notes-df package: Style files for biblatex

David Fussner    Version 0.9a (beta)  
[djf027@googlemail.com](mailto:djf027@googlemail.com)  
March 20, 2010

<b>Contents</b>		5.2 Citation Commands . . . . .	39
<b>1 Notice</b>	<b>1</b>	<b>6 Package Options</b>	<b>40</b>
<b>2 Quickstart</b>	<b>1</b>	6.1 Pre-Set biblatex Options . . . . .	40
2.1 License . . . . .	3	6.2 Biblatex-chicago-notes-df Options . . . . .	41
2.2 Acknowledgements . . . . .	3	<b>7 General Usage Hints</b>	<b>43</b>
<b>3 Detailed Introduction</b>	<b>4</b>	7.1 Loading the Package . . . . .	43
<b>4 The Specification</b>	<b>5</b>	7.2 Other Hints . . . . .	43
4.1 Entry Types . . . . .	5	<b>8 Internationalization</b>	<b>44</b>
4.2 Entry Fields . . . . .	16	<b>9 Interaction with Other Packages</b>	<b>46</b>
<b>5 Commands</b>	<b>35</b>	<b>10 Known Bugs</b>	<b>46</b>
5.1 Formatting Commands . . . . .	35	<b>11 Revision History</b>	<b>47</b>

## 1 Notice

Please be advised that this package is beta software, in at least two senses. First, it is merely a style to be used in conjunction with the `biblatex` package, whose author (Philipp Lehman) has made it clear that some parts of it may be subject to alteration before a final release. Any such changes may necessitate alterations to `biblatex-chicago-notes-df`, as well. Second, I have tried to implement as much of the *Chicago Manual of Style's* specification as seemed necessary for specialists in the humanities, though others may well disagree with me on these boundaries (especially with regard to legal citations, for example). Consider this beta release, then, a request for comments. If it seems like this package could be of use to you, yet it doesn't do something you need/want it to do, please feel free to let me know, and of course any suggestions for solving problems more elegantly or accurately would be most welcome.

## 2 Quickstart

**Important Note:** If you have used `biblatex-chicago-notes-df` before, please make sure you have read the `RELEASE` file that came with the package. It details the changes you'll need to make to your `.bib` database in order for it to work properly with this release. If you are new to this style, please read on.

The `biblatex-chicago-notes-df` package is designed for writers in the humanities who wish to use `LATEX` and `biblatex`, and who either want or need to format their references according to one of the standard humanities specifications. This package **does not** (yet) include any means of using the *Chicago Manual of Style's* "author-date" system, favored by many disciplines, but also, I believe, somewhat easier to implement as a minor modification to existing author-date styles, which include only a short reference in the text and a complete one in a list of references at the end. `Biblatex-chicago-notes-df`, by contrast, attempts to implement the *Manual's* "notes & bibliography" style, which includes a full reference in a first footnote, shorter references in subsequent notes, and a full reference in the bibliography. Some authors prefer to use the shorter note form even for the first occurrence, relying on the bibliography to provide the full information. This, too, is supported by the package.

Here's a list of things you will need in order to use `biblatex-chicago-notes-df`:

- Philipp Lehman's `biblatex` package, of course! You *must* use the latest version — 0.9 at time of writing — as that version introduces incompatibilities with previous releases, meaning that these style files will not function properly with earlier iterations of `biblatex`. Lehman's tools require several packages, and he strongly recommends several more:
    - `e-TEX` (required)
    - `etoolbox` — available from CTAN (required)
    - `keyval` — a standard package (required)
    - `ifthen` — a standard package (required)
    - `url` — a standard package (required)
    - `babel` — a standard package (*strongly* recommended)
    - `csquotes` — available from CTAN (recommended). Please upgrade to the latest version of `csquotes` (4.4d).
    - `bibtex8` — a replacement for `BITEX`, which can, with the right command-line switches, process very large `.bib` files. It also does the right thing when alphabetizing non-ASCII entries. It is available from CTAN (highly recommended).
    - `Biber` — alternatively, you can use Lehman's next-generation `BITEX` replacement, called `Biber`, which is available from SourceForge. You need the latest version (0.5.1) to work with `biblatex` 0.9, and it is required for users who are processing a `.bib` file in Unicode.
  - The line `\usepackage{biblatex-chicago}` in your document preamble. This takes care of loading `biblatex` with the `style=chicago-notes-df` option for you. Any other options you usually pass to `biblatex` can be given to `biblatex-chicago` instead, but loading it this way sets up a number of other parameters automatically. You can also load the package with the usual `\usepackage{biblatex}`, adding `style=chicagonotes-df`, but this is mainly for those who wish to set much of the low-level formatting of their document themselves. Please see section 7.1 below for a fuller discussion of the issues involved here.
  - If you wish to use the short note format even in the first reference, add `short` to the options of `\usepackage{biblatex-chicago}`, letting the bibliography provide the full reference.
  - If you are accustomed to using the `natbib` compatibility option with `biblatex`, then you can continue to do so with `biblatex-chicago`. If you are using `\usepackage{biblatex-chicago}` to load the package, then the option must be the plain `natbib` rather than `natbib=true`. If you use the latter, you'll get a `keyval` error. Please see section 6.2, below.
- New!
- By far the simplest setup is to use `babel`, and to have `american` as the main text language. As before, `babel-less` setups, and also those choosing `english` as the main text language, should work out of the box. This release also now provides support for German and French. Please see below (section 8) for a fuller explanation of all the options.
- New!
- The `chicago-notes-df.bbx`, `chicago-notes-df.cbx`, `biblatex-chicago.sty`, `cms-american.lbx`, `cms-french.lbx`, `cms-german.lbx`, and `cms-ngerman.lbx` files from `biblatex-chicago-notes-df`, installed either in a system-wide `TEX` directory, or in the working directory where you keep your `*.tex` files. If you use a system-wide directory, I would recommend first creating the `<TEXMFLOCAL>/tex/latex/biblatex-contrib/biblatex-chicago` directory, where `<TEXMFLOCAL>` is the root of your local `TEX` installation — for example, and depending on your system, `/usr/share/texmf-local`, `/usr/local/share/`

texmf, or C:\Local TeX Files\. Then you can place all seven of these files there, as it will be easier to update if they're all in one place. Of course, wherever you choose to place them in the texmf tree, you'll need to update the file name database to make sure TeX can find them.

- If you are using the XeLaTeX engine, please upgrade to the latest version of csquotes, which incorporates changes that should address formatting bugs that many users have encountered. Please see section 9 below for the details.
- Philipp Lehman's very clear and detailed documentation of the biblatex system, available in his package as biblatex.pdf. Here he explains why you might want to use the system, the rules for constructing .bib files for it, and the (numerous) methods at your disposal for modifying the formatted output.
- The annotated bibliography file chicago-test.bib, which will acquaint you with most of the details on how to get started constructing your own .bib file for use with biblatex-chicago-notes-df.
- The file sample.pdf, which shows how my system processes chicago-test.bib and sample.tex, in both footnotes and bibliography.
- The file you are reading, biblatex-chicago-notes-df.pdf, which aims to be as complete a description as possible of the rules for creating a .bib file that will, when processed by LaTeX and BibTeX, at least somewhat ease the burden when you try to implement the *Chicago Manual of Style's* specifications. If you have used a previous version of this package, please pay particular attention to the sections on Obsolete and Deprecated Features, starting on page 47. Please also note that, if you currently have quoted material in your .bib file, and are using \enquote or the standard LaTeX mechanisms there, then the simplest procedure is always to use \mkbibquote instead in order to ensure that punctuation works out right. (If you wish to place the four previous files in a system-wide directory, I would recommend <TEXMFLOCAL>/doc/latex/biblatex-contrib/biblatex-chicago, remembering of course to update the file name database afterward.)
- Access to a copy of *The Chicago Manual of Style* itself, which naturally contains incomparably more information than I can hope to present here. It should always be your first port of call when any doubts arise as to exactly what the specification requires.

## 2.1 License

Copyright © 2008–2010 David Fussner. This package is author-maintained. This work may be copied, distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in <http://www.latex-project.org/lppl.txt> and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later. This software is provided "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

## 2.2 Acknowledgements

Even a cursory glance at the chicago-notes-df.cbx and chicago-notes-df.bbx files will demonstrate how much of Lehman's code from biblatex I've adapted and re-used, and I've also followed some of the advice he gave to others in the comp.text.tex newsgroup. He has been instrumental in improving the contextual capitalization procedures of which the style makes such frequent use, and there are some other improvements he has recommended but which I shall implement in a future release. The code for formatting the footnote

marks, and that for printing the separating rule only after a run-on note, I've adapted from the `footmisc` package by Robin Fairbairns. I borrowed ideas for the `shorthandbibid` option from Dominik Waßenhoven's `biblatex-dw` package. There may be other  $\LaTeX$  code I've appropriated and forgotten, in which case please remind me. Finally, Charles Schaum and Joseph Reagle Jr. were both extremely generous with their help and advice during the development of this package, and have both continued indefatigably to test it and suggest needed improvements. They were particularly instrumental in encouraging the greatest possible degree of compatibility with other `biblatex` styles. Indeed, if the task of adapting `.bib` files for use with the Chicago style seems onerous now, you should have tried it before they got their hands on it.

### 3 Detailed Introduction

The *Chicago Manual of Style*, now in its 15th edition, has long, in America at least, been one of the most influential style guides for writers and publishers. While one's choices are now perhaps more extensive than ever, the *Manual* at least still provides a widely-recognized, and widely-utilized, standard. Personally, in humanities publications, I like having full references in footnotes, leaving the main body of text relatively unencumbered by the bibliographical apparatus, so from that point of view the Chicago style seems a perfect match. Indeed, when you add to this the sheer completeness of the specification, its detailed instructions for referencing an enormous number of different kinds of source material, then your choice (or your publisher's choice) of the *Manual* as a style guide seems set to be a happy one.

These very strengths, however, also make the style difficult to use. Admittedly, the *Manual* does leave room for "inventive solutions" to particular problems (17.2), and it also emphasizes consistency within a work, as opposed to rigid adherence to the specification, at least when writer and publisher agree (17.18). Sometimes a publisher demands such adherence, however, and anyone who has attempted to produce it may well come away with the impression that the specification itself is somewhat idiosyncratic in its complexity, and I can't help but agree. The numerous differences in punctuation (and strings identifying translators, editors, and the like) between footnotes and bibliographies, the sometimes unusual location of page numbers, the distinction between "journal" and "magazine," and the formatting differences between (e.g.) a work from antiquity and one from the Renaissance, all of these tend to overburden the writer who wants to comply with the standard. Many of these complexities, in truth, make the specification very nearly impossible to implement straightforwardly in a system like `BibTeX` — options multiply, each requiring a particular sort of formatting, until one almost reaches the point of believing that every individual book or article should have its own entry type. Completeness and usability tend each to exclude the other, so the code you have before you is a first attempt to achieve the former without utterly sacrificing the latter.

#### What `biblatex-chicago-notes-df` can and can't do

In short, the `biblatex` style files in this package try to simplify the task of following the Chicago "notes & bibliography" specification. The two sorts of reference are treated separately (as are the two different note forms, long and short), punctuation is placed within quotation marks when needed, and as a general rule as many parts of the style as possible are implemented as transparently as possible. Thanks to advice I received from Joseph Reagle Jr. and Charles Schaum while these files were a work in progress, I have attended as carefully as possible to backward compatibility with the standard `biblatex` styles, and have attempted to minimize any changes you need to make to achieve compliance with the Chicago specification. There is no doubt room for improvement on this score, but even now, for a substantial number of entries, any well-constructed `.bib` file that works for other `biblatex` styles will "just work"

under `biblatex-chicago-notes-df`. By no means, however, will all entries in such a `.bib` file produce equally satisfactory results. Using this documentation and the examples in `chicago-test.bib`, it should be possible to achieve compliance, though the amount of revision necessary to do so will vary significantly from `.bib` file to `.bib` file. Conversely, once you have created a database for `biblatex-chicago-notes-df`, it won't necessarily work well with other `biblatex` styles. Indeed, most, quite possibly all, users will find that they need to use special formatting macros within the `.bib` file that would make such a file unusable in any other context. I strongly recommend, if you want to experiment with this style, that you work on a copy of any `.bib` files that are important to you, until you have determined that this package does what you need/want it to do.

When I first began working on this package, I made the decision to alter as little as possible the main files from Lehman's `biblatex`, so that my `.bbx` and `.cbx` files would use his original `LATEX .sty` file and `BBTEX .bst` file. As you proceed, you will no doubt encounter some of the consequences of this decision, with certain fields and entry types in the `.bib` file having less-than-memorable names because I chose to use the supplementary ones provided by `biblatex.bst` rather than alter that file. I intended then, if it turned out that anyone besides myself actually used `biblatex-chicago-notes-df`, to ask Mr. Lehman to include more descriptive names for these few entry types and fields in `biblatex.bst`, if he were willing, but it may well be that retaining some of these non-descriptive names at least keeps them out of the way of the standard styles, somewhat enhancing compatibility between `.bib` files designed for Chicago and those designed for other specifications. As it turns out, several new types have appeared in `biblatex 0.8`, many of which I have incorporated as aliases to the custom entry types I defined before. If a consensus emerges about how best to assign the data to various fields in such entries, then I shall adopt it, but I intend to retain the old custom types as long as possible to enhance backward compatibility for Chicago users. Needless to say, I'm open to advice and suggestions on this score.

## 4 The Specification

In what follows, I attempt to explain all the parts of `biblatex-chicago-notes-df` that might be considered somehow "non standard," at least with respect to the styles included with `biblatex` itself, though in the section on entry fields I have also duplicated a lot of the information in `biblatex.pdf`, which I hope won't badly annoy expert users of the system. Headings in green indicate material new to this release, or occasionally old material that has undergone significant revision. Numbers in parentheses refer to sections of the *Chicago Manual of Style*, 15th edition. The file `chicago-test.bib` contains many examples from the *Manual* which, when processed using `biblatex-chicago-notes-df`, should produce the same output as you see in the *Manual* itself, or at least compliant output, where the specifications are vague or open to interpretation, a state of affairs which does sometimes occur. I have provided `sample.pdf`, which shows how my system processes `chicago-test.bib`, and I have also included the reference keys from the latter file below in parentheses.

New in this  
release

### 4.1 Entry Types

The complete list of entry types currently available in `biblatex-chicago-notes-df`, minus the odd `biblatex` alias, is as follows: **article**, **artwork**, **book**, **bookinbook**, **booklet**, **collection**, **customa**, **customb**, **customc**, **image**, **inbook**, **incollection**, **inproceedings**, **inreference**, **letter**, **manual**, **misc**, **online** (with its alias **www**), **patent**, **periodical**, **proceedings**, **reference**, **report** (with its alias **techreport**), **review**, **suppbook**, **suppcollection**, **suppperiodical**, **thesis** (with its aliases **mastersthesis** and **phdthesis**), and **unpublished**.

What follows is an attempt to specify all the differences between these types and the standard provided by `biblatex`. If an entry type isn't discussed here, then it is

safe to assume that it works as it does in the standard styles. In general, I have attempted not to discuss specific entry fields here, unless such a field is crucial to the overall operation of a given entry type. As a general and important rule, most entry types require very few fields when you use `biblatex-chicago-notes-df`, so it seemed to me better to gather information pertaining to fields in the next section.

**article** The *Chicago Manual of Style* (17.148) recognizes three different sorts of periodical publication, “journals,” “magazines,” and “newspapers.” The first (17.150) includes “scholarly or professional periodicals available mainly by subscription,” while the second refers to “weekly or monthly” publications that are “available either by subscription or in individual issues at bookstores or newsstands.” “Magazines” will tend to be “more accessible to general readers,” and typically won’t have a volume number. Indeed, by fiat I declare that should you need to refer to a journal that identifies its issues mainly by year, month, or week, then for the purposes of `biblatex-chicago-notes-df` such a publication is a “magazine,” and not a “journal.”

Now, for articles in “journals” you can simply use the traditional `BIBTEX` — and indeed `biblatex` — `article` entry type, which will work as expected and set off the page numbers with a colon, as required by the *Manual*. If, however, you need to refer to a “magazine” or a “newspaper,” then you need to add an `entrysubtype` field containing the exact string `magazine`. The main formatting differences between a magazine (which includes both “magazines” and “newspapers”) and a plain article are that the year isn’t placed within parentheses, and that page numbers are set off by a comma rather than a colon. Otherwise, the two sorts of reference have much in common. (For `article`, see *Manual* 17.154–181; `batson`, `beattie:crime`, `friedman:learning`, `garaud:gatine`, `garrett`, `hlatky:hrt`, `kern`, `lewis`, `loften:hamlet`, `mcmillen:antebellum`, `warr:ellison`, `white:callimachus`. For `entrysubtype magazine`, see 17.166, 17.182–198; `assocpress:gun`, `morgenson:market`, `reaves:rosen`, `rozner:liberation`, `stenger:privacy`.)

It gets worse. The *Manual* treats reviews (of books, plays, performances, etc.) as a sort of recognizable subset of “journals,” “magazines,” and “newspapers,” distinguished mainly by the way one formats the title of the review itself. In `biblatex` 0.7, happily, Lehman provided a review entry type which will handle a large subset of such citations, though not all. The key rule is this: if a review has a separate, non-generic title (`gibbard`; `osborne:poison`) in addition to something that reads like “review of ...,” then you need an `article` entry, with or without the `magazine entrysubtype`, depending on the sort of publication containing the review. If the only title is the generic “review of ...,” for example, then you’ll need the `review` entry type, with or without this same `entrysubtype` toggle using `magazine`. On review entries, see below. (The curious reader will no doubt notice that the code for formatting any sort of review still exists in `article`, as it was initially designed for `biblatex` 0.6, but this new arrangement is somewhat simpler and therefore, I hope, better.)

In the case of a review with a specific as well as a generic title, the former goes in the `title` field, and the latter in the `titleaddon` field. Standard `biblatex` intends this field for use with additions to titles that may need to be formatted differently from the titles themselves, and `biblatex-chicago-notes-df` uses it in just this way, with the additional wrinkle that it can, if needed, replace the title entirely, and this in, effectively, any entry type, providing a fairly powerful, if somewhat complicated, tool for getting `BIBTEX` to do what you want. Here, however, if all you need is a `titleaddon`, then you want to switch to the `review` type, where you can simply use the `title` field instead.

No less than six more things need explication here. First, since the *Manual* specifies that much of what goes into a `titleaddon` field stays unformatted — no italics, no quotation marks — this plain style is the default for such text, which means that you’ll have to format any titles within `titleaddon` yourself,

e.g., with `\mkbibemph{}`. Second, the *Manual* specifies a similar plain style for the titles of other sorts of material found in “magazines” and “newspapers,” e.g., obituaries, letters to the editor, interviews, the names of regular columns, and the like. References may contain both the title of an individual article and the name of the regular column, in which case the former should go, as usual, in a title field, and the latter in `titleaddon`. As with reviews proper, if there is only the generic title, then you want the review entry type. (See 17.188, 17.190, 17.193; `morgenson:market`, `reaves:rosen`.)

Third, the *Manual* suggests that, in the case of “unsigned newspaper articles or features . . . the name of the newspaper stands in place of the author” (17.192). It doesn’t always carry through on this in its own presentation of newspaper citations (see esp. 17.188), but I’ve implemented their recommendation nonetheless, which means that in an article entry, `entrysubtype magazine`, or in a review entry, `entrysubtype magazine`, and only in such entries, a missing author field results in the name of the periodical (in the `journaltitle` field) being used as the missing author. If, for reasons of emphasis or merely because of personal preference, you wish to keep the title in initial position, then you need to define the author using, as you can see from the examples in `chicago-test.bib`, the `biblatex` macro `\isdot`, which functions, in this context, to define the author, yet to print nothing. Note that if you choose to use the name of the newspaper as an author, then you’ll need to define the `shortauthor` field (with formatting) in order to make the short footnote form work properly, and you’ll also need a `sortkey` field to ensure that the bibliography entry is alphabetized by `journaltitle` rather than by title. (See `lakeforester:pushcarts`.)

Fourth, if you’ve been using `biblatex-chicago-notes-df` for a while, you’ll no doubt be accustomed to using the single-letter `\bibstring` mechanism in order to help `biblatex` decide where to capitalize a wide variety of strings in numerous entry fields. This mechanism was particularly common in all the periodical types, but if you’ve had a look in `chicago-test.bib` while following this documentation, you’ll have noticed that it no longer appears there. The regular whole-word `bibstrings` still work as normal, but the single-letter ones are now obsolete, replaced by Lehman’s macro `\autocap`, which itself only occurs twice in `chicago-test.bib`. Basically, in certain fields, just beginning your data with a lowercase letter activates the mechanism for capitalizing that letter depending on its context within a note or bibliography entry. Please see `\autocap` below for the details, but both the `titleaddon` and `note` fields are among those treating their data this way, and since both appear regularly in article entries, I thought the problem merited a preliminary mention here.

Fifth, if you need to cite an entire issue of any sort of periodical, rather than one article in an issue, then the `periodical` entry type, once again with or without the `magazine` toggle in `entrysubtype`, is what you’ll need. (You can also use the `article` type, placing what would normally be the `issuetitle` in the title field and retaining the usual `journaltitle` field, but this arrangement isn’t compatible with standard `biblatex`.) The `note` field is where you place something like “special issue” (with the small “s” enabling the automatic capitalization routines), whether you are citing one article or the whole issue (`conley:fifthgrade`, `good:wholeissue`). Indeed, this is a somewhat specialized use of `note`, and if you have other sorts of information you need to include in an article, `periodical`, or review entry, then you shouldn’t put it in the `note` field, but rather in `titleaddon` or perhaps `addendum` (`brown:bremmer`).

Finally, and in the interests of completeness, it may be as well to suggest that if you wish to cite a television or radio broadcast, the `article` type, `entrysubtype magazine` is the place for it. The name of the program would go in `journaltitle`, with the name of the episode in `title`. The network’s name now goes into the new `usera` field, replacing the formatting kludge I suggested in version 0.7. Of course, if the piece you are citing has only a generic name (an interview, for

example), then the review type would be the best place for it. (8.196, 17.207; see bundy:macneil for an example of how this all might look in a .bib file.)

If you're still with me, allow me to recommend that you browse through `chicago-test.bib` to get a feel for just how many of the *Manual's* complexities the article and review (and, indeed, periodical) types attempt to address. It may be that in future releases of `bibtex-chicago-notes-df` I'll be able to simplify these procedures somewhat, but in the meantime it might be of some comfort that I have found in my own research that the unusual and/or limit cases are really rather rare, and that the vast majority of sources won't require any knowledge of these onerous details.

**artwork** Arne Kjell Vikhagen has pointed out to me that none of the standard entry types were straightforwardly adaptable when referring to visual artworks. The *Manual* doesn't give any thorough specifications for such references, and indeed it's unclear that it believes it necessary to include them in the bibliographical apparatus at all. Still, it's easy to conceive of contexts in which a list of artworks studied might be desirable, and `bibtex` includes entry types for just this purpose, though the standard styles leave them undefined. The two I have included in this release are `artwork` and `image`, the former intended for paintings, sculptures, etchings, and the like, the latter for photographs. The two entry types work in exactly the same way as far as constructing your .bib entry, and when printed the only difference will be that the titles of artworks are italicized, those of images placed within quotation marks.

As one might expect, the artist goes in author and the name of the work in title. The `type` field is intended for the medium — e.g., oil on canvas, charcoal on paper — and the `version` field might contain the state of an etching. You can place the dimensions of the work in note, and the current location in organization, institution, and/or location, in ascending order of generality. The `type` field, as in several other entry types, uses `bibtex's` automatic capitalization routines, so if the first word only needs a capital letter at the beginning of a sentence, use lowercase in the .bib file and let `bibtex` handle it for you. (See *Manual* 12.33; leo:madonna, bedford:photo.)

As a final complication, the *Manual* (8.206) says that “the names of works of antiquity . . . are usually set in roman.” If you should need to include such a work in the reference apparatus, you can either define an `entrysubtype` for an artwork entry — anything will do — or you could use the `misc` entry type with an `entrysubtype`. Fortunately, in this instance the other fields in a `misc` entry function pretty much as in `artwork` or `image`.

**bookinbook** With this release of `bibtex`, Lehman has added the `bookinbook` entry type for referring to parts of books that are considered, in other contexts, themselves to be books, rather than chapters, essays, or articles. The `customb` entry type has up to now provided this functionality for `bibtex-chicago-notes-df`, and for the moment at least you can use these two types completely interchangeably. Please see the documentation of `customb` below for how to construct your bib file entry.

**booklet** This is the first of two entry types — the other being `manual`, on which see below — which are traditional in `BTEX` styles, but which the *Manual* (17.241) suggests may well be treated basically as books. In the interests of backward compatibility, `bibtex-chicago-notes-df` will so format such an entry, which uses the `howpublished` field instead of a standard publisher, though of course if you do decide just to use a `book` entry then any information you might have given in a `howpublished` field should instead go in publisher. (See clark:mesopot.)

**customa** This is the entry type to use for citing letters, memoranda, or similar texts, but *only* when they appear in a published collection. (Unpublished material of this nature needs a `misc` entry, for which see below.) Depending on what sort of



information you need to present in a citation, you may simply be able to get away with a standard book entry, which may then be cited by page number (see *Manual* 17.31, 17.42; meredith:letters, adorno:benj). If, however, for whatever reason, you need to give full details of a specific letter, then you'll need to use the `customa` entry type, which attempts to simplify for you the *Manual's* rather complicated rules for formatting such references. (See 17.76–78; jackson:paulina:letter, white:ross:memo, white:russ [a completely fictitious entry to show the `xref` mechanism], white:total [a book entry, for the bibliography]).

New!

To start, the name of the letter writer goes in the `author` field, while the `title` field contains both the name of the writer and that of the recipient, in the form `Author to Recipient`. The `titleaddon` field contains the type of correspondence involved. If it's a letter, this field may be left blank, but if it's a memorandum or report or the like, then this is the place to specify that fact. Also, because the `origdate` field only accepts numbers, if you want to use the abbreviation "n.d." (or `\bibstring{nodate}`) for undated letters, then this is where you should put it. Most importantly, and this is new for this release, the date of the letter itself goes in the `origdate` field (year-month-day), which now allows a full date specification, while the publishing date of the whole collection goes in the `date` field, instead of in the obsolete `origyear`. As in other entry types, then, the `date` field now has its ordinary meaning of "date of publication." (You may have noticed here that the presentation of the `origdate` in this sort of reference is different from the date format required elsewhere by the *Manual*. This appears to result from some recent changes to the specification, and it may be that we could get away with choosing one or the other format for all occurrences [6.46], but for the moment I hope this mixed solution will suffice.) Another difficulty arises when producing the short footnote form, which requires you to provide a `shorttitle` field of the form "to Recipient," the latter name as short as possible while avoiding ambiguity. The remaining fields are fairly self explanatory, but do remember that the title of the published collection belongs in `booktitle` rather than in `title`.

Finally, the *Manual* specifies that if you cite more than one letter from a given published collection, then the bibliography should contain only a reference to said collection, rather than to each individual letter, while the form of footnotes would remain the same. This should be possible using `BTBTEX's` standard `crossref` field, with each `customa` entry pointing to a collection or book entry, for example. I shall discuss cross references at length later (`crossref` and `xref`, below), but I should mention here that `customa` is one of the entry types in which a `crossref` or an `xref` field automatically results in special shortened forms in notes and bibliography if more than one piece from a single collection is cited. (The other entry types are `incollection` and `inproceedings`; see 17.70 for the *Manual's* specification.) This ordinarily won't be an issue for `customa` entries in the bibliography, as individual letters aren't included there, but it is operative in notes, where you can disable it simply by not using a `crossref` or an `xref` field. In the `crossref` docs, below, I recommend a way of keeping the individual letters from turning up in the bibliography, involving the use of the `keywords` field.

**NB:** The standard `biblatex` entry type `letter` currently functions as an alias for `customa`.

customb

I initially provided this non-standard entry type because I believed it necessary to include some means of referring to parts of books that are considered, in other contexts, themselves to be books, rather than chapters, essays, or articles. Version 0.9 of `biblatex` now provides `bookinbook` for this purpose, and for the moment you can continue using either entry type interchangeably. Such an entry can have a `title` and a `maintitle`, but it can also contain a `booktitle`, all three of which will be italicized in the reference matter. In general usage it is, therefore, rather like the traditional `inbook` type, only with its title in italics rather than in quotation marks. (See *Manual* 17.72, 17.89, 17.93; bernard:boris, euripides:orestes, plato:republic:gr.)

**NB:** The Euripides play receives slightly different presentations in 17.89 and 17.93. Although the specification is very detailed, it doesn't eliminate all choice or variation. Using a system like BibTeX should help to maintain consistency.

**customc** This is the entry type to use if the main focus of a reference is supplemental material in a book or in a collection, e.g., an introduction, afterword, or forward, either by the same or a different author. In previous releases of `biblatex-chicago-notes-df` these three just-mentioned types of material, and only these three types, could be referenced using the `introduction`, `afterword`, or `foreword` fields, a system that required you simply to define one of them in any way and leave the others undefined. The macros don't use the text provided by such an entry, they merely check to see if one of them is defined, in order to decide which sort of pre- or post-matter is at stake, and to print the appropriate string before the title in long notes, short notes, list of shorthands, and bibliography. I have retained this mechanism both for backward compatibility and because it works without modification across multiple languages, but have also added functionality which allows you to cite any sort of supplemental material whatever, using the `type` field. Under this system, simply put the nature of the material, including the relevant preposition, in that field, beginning with a lowercase letter so `biblatex` can decide whether it needs capitalization depending on the context. Examples might be "preface to" or "colophon of." (Please note, however, that unless you use a `\bibstring` command in the `type` field, the resultant entry will not be portable across languages.)

The other rules for constructing your `.bib` entry remain the same. The `author` field refers to the author of the introduction or afterword, while `bookauthor` refers to the author of the main text of the work, if the two differ. If the focus of the reference is the main text of the book, but you want to mention the name of the writer of an introduction or afterword for bibliographical completeness, then the normal `biblatex` rules apply, and you can just put their name in the appropriate field of a book entry, that is, in the `foreword`, `afterword`, or `introduction` field.

*suppbook*  
*suppcollection* Please note also that `biblatex-chicago-notes-df` incorporates the new `biblatex` entry types very kindly provided by Lehman in version 0.8 to facilitate references of this sort, to wit, `suppbook` and `suppcollection`. Both of them currently function as aliases to `customc`, but it is conceivable in the future that the latter, less descriptive entry type might be phased out in favor of the standard `biblatex` types. Indeed, if a consensus emerges across `biblatex` styles for which fields to use for all of this data, then the two new entry types will cease to be aliases and adopt that consensus. For the moment, I have thought it important to retain backward compatibility, but you should be aware that evolving standards may impose changes in order to improve cross compatibility among `biblatex` styles. (See *Manual* 17.74–75; `polakow:afterw`, `prose:intro`).

**image** This entry type, left undefined in the standard styles, is in `biblatex-chicago-notes-df` intended for referring to photographs. Excluding the possible use of the `entrysubtype` field, which in an `image` entry would be ignored, this type is a clone of `artwork`, so you should consult the latter's documentation above to see how to construct your `.bib` entry. (See *Manual* 12.33; `bedford:photo`.)

**inbook**  
**inccollection** These two standard `biblatex` types have very nearly identical formatting requirements as far as the Chicago specification is concerned, but I have retained both of them for compatibility. `Biblatex.pdf` (§ 2.1.1) intends the first for "a part of a book which forms a self-contained unit with its own title," while the second would hold "a contribution to a collection which forms a self-contained unit with a distinct author and its own title." The title of both sorts will be placed within quotation marks, and in general you can use either type for most material falling into these categories. There is, however, an important difference between them, as it is only in `inccollection` entries that I implement the

*Manual's* recommendations for space-saving abbreviations in notes and bibliography when you cite multiple pieces from the same collection. These abbreviations are activated when you use the `crossref` or `xref` field in `incollection` entries, and not in `inbook` entries, mainly because the *Manual* (17.70) here specifies a “multiauthor book.” (For more on this mechanism see `crossref`, below, and note that it is also active in `customa` and `inproceedings` entries.) If the part of a book to which you are referring has had a separate publishing history as a book in its own right, then you may wish to use the `customb` type, instead, on which see above. (See *Manual* 17.68–72; `inbook: ashbrook:brain, phibbs:diary, will:cohere; incollection: centinel:letters, contrib:contrib, sirosh:visualcortex; ellet:galena, keating:dearborn, and lippincott:chicago` [and the collection entry `prairie:state`] demonstrate the use of the `crossref` field with its attendant abbreviations in notes and bibliography.)

**NB:** The *Manual* suggests that, when referring to a chapter, one use either a chapter number or the inclusive page numbers, not both. If, however, you wish to refer in a footnote to a specific page within the chapter, `biblatex-chicago-notes-df` will always print the optional, postnote argument of a `\cite` command — the page number, say — instead of any chapter number or inclusive page numbers given in the `.bib` file `incollection` entry. This mechanism is quite general, that is, any specific page reference given in any sort of `\cite` command overrides the contents of a `pages` field in a `.bib` file entry.

Currently, only the `chapter` field receives this treatment in `biblatex-chicago-notes-df`, though it may be possible in the future to generalize this to other named parts of a book. Do let me know if this would be helpful to you.

**inproceedings** This entry type works pretty much as in standard `biblatex`. Indeed, the main differences between it and `incollection` are the lack of an `edition` field and the possibility that an organization may be cited alongside the publisher, even though the *Manual* doesn't specify its use (17.71). Please note, also, that the `crossref` and `xref` mechanism for shortening citations of multiple pieces from the same proceedings is operative here, just as it is in `incollection` entries. See `crossref`, below, for more details.

**inreference** This entry type is aliased to `incollection` in the standard styles, but the *Manual* has particular requirements, so if you are citing “[w]ell-known reference books, such as major dictionaries and encyclopedias,” then this type should simplify the task of conforming to the specifications (17.238–239). The main thing to keep in mind is that I have designed this entry type for “alphabetically arranged” works, which you shouldn't cite by page, but rather by the name(s) of the article(s). Because of the formatting required by the *Manual*, we need one of `biblatex's` list fields for this purpose, and in order to keep all this out of the way of the standard styles, I have chosen the `lista` field. You should present these article names just as they appear in the work, separated by the keyword “and” if there is more than one, and `biblatex-chicago-notes-df` will provide the appropriate prefatory string (`s.v.`, plural `s.vv.`), and enclose each in its own set of quotation marks (`ency:britannica`). In a typical `inreference` entry, very few other fields are needed, as “the facts of publication are often omitted, but the edition (if not the first) must be specified.” In practice, this means a title and possibly an `edition` field.

There are quite a few other peculiarities to explain here. First of all, you should present any well-known works *only* in notes, not in a bibliography, as your readers are assumed to know where to go for such a reference. This requires the keywords mechanism I discuss below under `crossref` and `keywords`. For such works, and given how little information will be present even in a full note, you may wish to use `\fullcite` or `\footfullcite` in place of the short form, especially if, for example, you are citing different versions of an article appearing in different editions.

If the work is slightly less well known, it may be that full publication details are appropriate (times:guide), but this makes things more complicated. In previous releases of `biblatex-chicago-notes-df`, you would have had to format the postnote field of short notes appropriately, including the prefatory string and quotation marks I mentioned above. Starting with this release you can put an article name in the postnote field of inreference entries and have it formatted for you, and this holds for both long and short notes, which could allow you to refer separately to many different articles from the same reference work using only one .bib entry. (In a long note, any postnote field stops the printing of the contents of lista.) The only limitation on this system is that the postnote field, unlike lista, is not a list, and therefore for the formatting to work correctly you can only put one article name in it. Despite this limitation, I hope that the current system might simplify things for users who cite numerous works of reference.

If it seems appropriate to include such a work in the bibliography, be aware that the contents of the lista field will also be presented there, which may not be what you want. A separate reference entry might solve this problem, but you may also need a sortkey field to ensure proper alphabetization, as biblatex will attempt to use an editor or author name, if either is present. (Cf. mla:style, a reference entry that uses section numbers instead of alphabetized headings, and useeditor=false in the options field instead of a sortkey to ensure the correct alphabetization.)

Speaking of the author, this field holds the author of the specific entry (in lista), not the author of the title as a whole. This name will be printed in parentheses after the entry's name (grove:sibelius). If you wish to refer to a reference work by author or indeed by editor, having either appear at the head of the note (long or short) or bibliography entry, then you'll need to use a book entry instead (cf. schellinger:novel), where the lista mechanism will also work, but which in every other way will be treated as a normal book, often a good choice for unfamiliar or non-standard reference works.

Finally, all of these rules apply to online reference works, as well, for which you need to provide not only a url but also, always, a urldate, as these sources are in constant flux (wikiped:bibtex, grove:sibelius).

**letter** This was a new entry type in biblatex 0.8, and is currently left undefined by the standard styles. Previous releases of `biblatex-chicago-notes-df` provided `customa` for letters or memoranda in published collections, and `misc` (with `entrysubtype`) for unpublished texts of the same sort. For the moment, the `letter` type is an alias for `customa`, so you should see its documentation for full details on how to construct your .bib entries for published letters. Please keep in mind, however, that if a general consensus evolves across biblatex styles concerning how to handle this data, then the syntax of `letter` entries might well change in order to maintain cross compatibility.

**manual** This is the second of two traditional BibT<sub>E</sub>X entry types that the *Manual* suggests formatting as books, the other being `booklet`. As with this latter, I have retained it in `biblatex-chicago-notes-df` for backward compatibility, its main peculiarity being that, in the absence of a named author, the organization producing the manual will be provided both as author and as publisher, and you'll need to provide a `shortauthor` for use in the short note form. (In such a case, you'll also need a `sortkey` field to aid biblatex's alphabetization routines.) Of course, if you were to use a `book` entry for such a reference, then you would need to define both `author` and `publisher` using the name you here might have put in `organization`. (See 17:47; `chicago>manual`, `dyna:browser`, `natrecoff:camera`.)

**misc** As its name suggests, the `misc` entry type was designed as a hold-all for citations that didn't quite fit into other categories. In `biblatex-chicago-notes-df`, I have somewhat extended its applicability, while retaining its traditional use. Put simply, with no `entrysubtype` field, a `misc` entry will retain backward com-

patibility with the standard styles, so the usual howpublished, version, and type fields are all available for specifying an otherwise unclassifiable text, and the title will be italicized. (The *Manual*, you may wish to note, doesn't give specific instructions on how such citations should be formatted, so when using the Chicago style I would recommend you have recourse to this traditional entry type as sparingly as possible.)

If you do provide an `entrysubtype` field, the `misc` type provides a means for citing unpublished letters, memoranda, private contracts, wills, interviews, and the like, making it something of an unpublished analogue to the `customa`, `article`, and `review` entry types (which see). Typically, such an entry will cite part of an archive, and equally typically the text cited won't have a specific title, but only a generic one, whereas an unpublished entry will ordinarily have a specific author and title, and won't come from a named archive. The `misc` type with an `entrysubtype` defined is the least formatted of all those specified by the *Manual*, so titles are in plain text, and any location details take no parentheses in full footnotes. If you are wondering what to put in `entrysubtype`, the answer is, in `biblatex-chicago-notes-df 0.9`, anything at all. You no longer need to put the exact string `letter` there in order to move the date into closer proximity with the title, rather, you just place the date of the letter (or perhaps a dated interview) into the `origdate` field rather than into `date`. If you don't wish to have the date so attached to the title, then use `date` instead of `origdate`. In effect, whether it's a `customa`, `letter`, or `misc` entry, it is by using the new `origdate` field that you identify when a letter (or anything like it) was written. This change, I hope, maintains consistency of usage across entry types and also reduces the complexity of using the `misc` type. A defined `entrysubtype` is still necessary, however, to ensure correct formatting of the title, even if the placement of the date is of no interest — any entry without an `entrysubtype` will be treated as a traditional `misc` entry, and the title italicized. In addition, defining `entrysubtype` activates the automatic capitalization mechanism in the title field of `misc` entries, on which see `\autocap` below. (See `17.205-206`, `17.220`, `17.222-232`; `creel:house`, `dinkel:agassiz`.)

New!

As in `customa` entries, the titles of unpublished letters are of the form `Author to Recipient`, and further information can be given in the `titleaddon` field, including the abbreviation "n.d." (or `\bibstring{nodate}`) for undated examples. The `note`, `organization`, `institution`, and `location` fields (in ascending order of generality) allow the specification of which manuscript collection now holds the letter, though the *Manual* specifies (17.228) that well-known depositories don't usually need a city, state or country specified. (The traditional `misc` fields are all still available, also.) The short note form can use the same title, but you may need to define the `shortauthor` field with, e.g., an `\isdot` command to make it work (`creel:house`). If you want to include the date of a letter in a short note, I have provided the `\letterdatelong` command for inclusion in the `postnote` field of the citation command.

As with `customa` entries, the *Manual* (17.223) suggests that bibliography entries contain only the name of the manuscript collection, unless only one item from that collection is cited. The `crossref` field can be used, as well as the `keywords` mechanism for preventing the individual items from turning up in the bibliography. Obviously, this is a matter for your discretion, and if you're using only short notes (see the `short` option, below), you may feel the need to include more information in the note if the bibliography doesn't contain a full reference to an individual item.

Finally, if the `misc` entry isn't a letter, remember that, as in `article` and `review` entries, words like `interview` or `memorandum` needn't be capitalized unless they follow a period — the automatic capitalization routines (with the `title` field starting with a lowercase letter [see `dinkel:agassiz`, `spock:interview`, and `\autocap`]) will ensure correctness. In all this class of archived material, the *Manual* (17.222) quite specifically requires more consistency within your own work than confor-

mity to some external standard, so it is the former which you should pursue. I hope that `bibtex-chicago-notes-df` proves helpful in this regard.

**online** The *Manual*'s instructions (17.142–147, 17.198, 17.234–237) for citing online materials are slightly different from those suggested by standard `bibtex`. Indeed, this is a case where complete backward compatibility with other `bibtex` styles may be impossible, because as a general rule the *Manual* considers relevant not only where a source is found, but also the nature of that source, e.g., if it's an online edition of a book (james:ambassadors), then it calls for a book entry. Even if you cite an "intrinsically online" source, if that source is structured more or less like a conventional printed periodical, then you'll probably want to use article or review instead of online (stenger:privacy, which cites *CNN.com* — *Yahoo! News* is another example that would be treated in such a way). If the "standard facts of publication" are missing, then the online type is usually the best choice (evanston:library, powell:email). Some online materials will, no doubt, make it difficult to choose an entry type, but so long as all locating information is present, then perhaps that is enough to fulfill the specification, or at least so I'd like to hope.

Constructing an online `.bib` file entry is much the same as in `bibtex`. The title field would contain the title of the page, the organization field could hold the title or owner of the whole site. If there is no specific title for a page, but only a generic one (powell:email), then such a title should go in `titleaddon`, not forgetting to begin that field with a lowercase letter so that capitalization will work out correctly.

**patent** The patent entry type has been thoroughly revised for this release, breaking backward compatibility with earlier versions of `bibtex-chicago-notes-df`. (The *Manual* is very brief on the subject [17.219], but very clear about which information it wants you to present, so such entries may not work well with other `bibtex` styles.) You should place the filing date in the new `origdate` field, and `bibtex-chicago-notes-df` will automatically prepend the bibstring `patentfiled` to that date. If the patent has been granted, then you put the date it was issued in the `date` field, to which the bibstring `patentissued` will automatically be prepended. In other words, you no longer need to use a hand-formatted addendum field, though you can place additional information in that field if needed, and it will be printed in close association with the dates. See `petroff:impurity` in `chicago-test.bib`. The patent number goes in the `number` field, and you should use the standard `bibtex` bibstrings in the `type` field. Though it isn't mentioned by the *Manual*, `bibtex-chicago-notes-df` will print the holder after the author, if you provide one.

**periodical** This is the standard `bibtex` entry type for presenting an entire issue of a periodical, rather than one article within it. It has the same function in `bibtex-chicago-notes-df`, and in the main uses the same fields, though in keeping with the system established in the article entry type (which see) you'll need to provide `entrysubtype magazine` if the periodical you are citing is a "newspaper" or "magazine" instead of a "journal." Also, remember that the `note` field is the place for identifying strings like "special issue," with its initial lowercase letter to activate the automatic capitalization routines. (See *Manual* 17.170; `good:wholeissue`.)

**reference** This entry type is aliased to `collection` by the standard `bibtex` styles, but I intend it to be used in cases where you need to cite a reference work but not an alphabetized entry or entries in that work. This could be because it doesn't contain such entries, or perhaps because you intend the citation to appear in a bibliography rather than in notes. Indeed, the only differences between it and `inference` are the lack of a `lista` field to present an alphabetized entry, and the fact that any `postnote` field will be printed verbatim, rather than formatted as an alphabetized entry. (See `mla:style` for an example of a reference work that uses numbered sections rather than alphabetized entries, and that appears in the bibliography as well.)

**report** This entry type is a biblatex generalization of the traditional `BIBTEX` type `techreport`. Instructions for such entries are rather thin on the ground in the *Manual* (17.241), so I have followed the generic advice about formatting it like a book, and hope that the results conform to the specification. Its main peculiarities are the `institution` field in place of a `publisher`, the `type` field for identifying the kind of report in question, and the `isrn` field containing the International Standard Technical Report Number of a technical report. As in standard biblatex, if you use a `techreport` entry, then the `type` field automatically defaults to `\bibstring{techreport}`. As with `booklet` and `manual`, you can also use a book entry, putting the report type in note and the institution in publisher. (See `herwign:office`.)

**review** The review entry type was added to biblatex 0.7, and it certainly eases the task of coping with the *Manual*'s complicated requirements for citing periodicals of all sorts, though it doesn't, I admit, eliminate all difficulties. As its name suggests, this entry type was designed for reviews published in periodicals, and if you've already read the article instructions above — if you haven't, I recommend doing so now — you'll know that `review` serves as well for citing other sorts of material with generic titles, like letters to the editor, obituaries, interviews, and the like. The primary rule is that any piece that has only a generic title, like "review of ...," "interview with ...," or "obituary of ...," calls for the review type. Any piece that also has a specific title, e.g., "'Lost in `BIBTEX`,' an interview with ...," requires an article entry. (This assumes the text is found in a periodical of some sort. Were it found in a book, then the `incollection` type would serve your needs, and you could use `title` and `titleaddon` there. While we're on the topic of exceptions, the *Manual* includes an example — 17.207 — where the "Interview" part of the title is considered a subtitle rather than a `titleaddon`, said part therefore being included inside the quotation marks and capitalized accordingly. Not having the journal in front of me I'm not sure what prompted that decision, but `biblatex-chicago-notes-df` would obviously have no difficulty coping with such a situation.)

Once you've decided to use `review`, then you need to determine which sort of periodical you are citing, the rules for which are the same as for an article entry. If it is a "magazine" or a "newspaper", then you need an `entrysubtype` `magazine`. The generic title goes in `title` and the other fields work just as they do in an article entry with the same `entrysubtype`, including the substitution of the `journaltitle` for the author if the latter is missing. (See 17.185, 17.188–194, 17.199–203, 17.207; `barcott:review`, `bundy:macneil`, `Clemens:letter`, `gourmet:052006`, `kozinn:review`, `nyt:obittrevor`, `nyt:trevorobit`, `unsigned:ranke`, `wallraff:word`.) If, on the other hand, the piece comes from a "journal," then you don't need an `entrysubtype`. The generic title goes in `title`, and the remaining fields work just as they do in a plain article entry. (See 17.201; `ratliff:review`.)

The onerous details are the same as I described them in the **article** section above, but I'll repeat some of them briefly here. If anything in the title needs formatting, you need to provide those instructions yourself, as the default is completely plain. In the short note form this holds as well for when the title of a periodical replaces the author, in which case you'll need to provide a formatted `shortauthor` field (`gourmet:052006`, `nyt:trevorobit`). If you wish to keep the title at the head of an entry, then you'll need to define `author` with `\isdot`, (as in `nyt:obittrevor`, by contrast with `nyt:trevorobit`). As in `misc` entries with an `entrysubtype`, words like "interview," "review," and "letter" only need capitalization after a full stop, i.e., ordinarily in a bibliography and not a note, so `biblatex-chicago-notes-df` automatically deals with this problem itself if you start the title field with a lowercase letter. The file `chicago-test.bib` and the documentation of `\autocap` will provide guidance here.

**suppbook** This entry type, new in biblatex 0.8, is intended to hold supplemental material from a book, such as an introduction or a preface. Previous versions of biblatex-

chicago-notes-df provided the `customc` type for this purpose, and now you can use either of these, as I've added `suppbook` as an alias of `customc`, though strictly speaking it should perhaps be the other way around. Please see above under **customc** for the full instructions on how to construct a `.bib` entry for such a reference, and note that `customc` now accommodates any sort of additional material through the use of the `type` field, recently added to `biblatex-chicago-notes-df`.

**suppcollection** This is also new to `biblatex` 0.8, and fulfills a function analogous to `suppbook`. Indeed, I believe the **customc** type can serve to present supplemental material in both types of work, so this entry type is, like `suppbook`, an alias to `customc`, which see.

**suppperiodical** This type, new to `biblatex` 0.8, is intended to allow reference to generically-titled works in periodicals, such as regular columns or letters to the editor. Previous releases of `biblatex-chicago-notes-df` provided the `review` type for this purpose, and now you can use either of these, as I've added `suppperiodical` as an alias of `review`. Please see above under **review** for the full instructions on how to construct a `.bib` entry for such a reference.

**unpublished** The unpublished entry type works largely as it does in standard `biblatex`, though it's worth remembering that you should use a lowercase letter at the start of your `note` field (or perhaps an `\autocap` command in the somewhat contradictory `howpublished`, if you have one) for material that wouldn't ordinarily be capitalized except at the beginning of a sentence (`nass:address`).

## 4.2 Entry Fields

The following discussion presents, in alphabetical order, a complete list of the entry fields you will need to use `biblatex-chicago-notes-df`. As in section 4.1, I shall include references to the numbered paragraphs of the *Chicago Manual of Style*, and also to the entries in `chicago-test.bib`. Many fields are most easily understood with reference to other, related fields. In such cases, cross references should allow you to find the information you need.

**addendum** As in standard `biblatex`, this field allows you to add miscellaneous information to the end of an entry, after publication data but before any `url` or `doi` field. In the `patent` entry type (which see), it will be printed in close association with the filing and issue dates. In any entry type, if your data begins with a word that would ordinarily only be capitalized at the beginning of a sentence, then simply ensure that that word is in lowercase, and the style will take care of the rest. Cf. `note`. (See *Manual* 17.145, 17.123; `davenport:attention`, `natrecoff:camera`.)

**afterword** In most circumstances, this field will function as it does in standard `biblatex`, i.e., you should include here the author(s) of an afterword to a given work. The *Manual* suggests that, as a general rule, the afterword would need to be of significant importance in its own right to require mentioning in the reference apparatus, but this is clearly a matter for the user's judgment. As in `biblatex`, if the name given here exactly matches that of an editor and/or a translator, then `biblatex-chicago-notes-df` will concatenate these fields in the formatted references.

As noted above, however, this field has a special meaning in the `customc` entry type, used to make an afterword, foreword, or introduction the main focus of a citation. If it's an afterword at issue, simply define `afterword` any way you please, leave `foreword` and `introduction` undefined, and `biblatex-chicago-notes-df` will do the rest. Cf. `foreword` and `introduction`. (See *Manual* 17.46, 17.74; `polakow:afterw`.)

**annotation** At the request of Emil Salim, `biblatex-chicago-notes-df` has, as of version 0.9, added a `package` option (see `annotation` below, section 6.2) to allow you to produce annotated bibliographies. The formatting of such a bibliography is



currently fairly basic, though it conforms with the *Manual*'s minimal guidelines (16.77). The default in `chicago-notes-df.cbx` is to define `\DeclareFieldFormat{annotation}` using `\par\nobreak\vskip\bibitemsep`, though you can alter it by re-declaring the format in your preamble. The page-breaking algorithms don't always give perfect results here, but the default formatting looks, to my eyes, fairly decent. In addition to tweaking the field formatting you can also insert `\par` (or even `\vadjust{\eject}`) commands into the text of your annotations to improve the appearance. Please consider the `annotation` option a work in progress, but it is usable now. (N.B.: The `BIBTEX` field `annotate` serves as an alias for this.)

**annotator** I have implemented this biblatex field pretty much as that package's standard styles do, even though the *Manual* doesn't actually mention it. It may be useful for some purposes. Cf. `commentator`.

**author** For the most part, I have implemented this field in a completely standard `BIBTEX` fashion. Remember that corporate or organizational authors need to have an extra set of curly braces around them (e.g., `{Associated Press}`) to prevent `BIBTEX` from treating one part of the name as a surname (17.47, 17.197; `assoc-press:gun`, `chicago:manual`). If there is no author, then `biblatex-chicago-notes-df` will look, in sequence, for an editor, translator, or compiler (actually `namec`, currently) and use that name (or those names) instead, followed by the appropriate identifying string (esp. 17.41, also 17.28–29, 17.88, 17.95, 17.172; `boxer:china`, `brown:bremer`, `harley:cartography`, `schellinger:novel`, `sechzer:women`, `silver:gawain`, `soltes:georgia`). Please note that when a `namec` appears at the head of an entry, you'll need to assist `biblatex`'s sorting algorithms by providing a `sortkey` field to ensure correct alphabetization in the bibliography. Also, a `shortauthor` entry is necessary to provide a name at the head of the short note form.

In the rare cases when this substitution mechanism isn't appropriate, you have two options: either you can (`chaucer:liferecords`) put all the information into a `note` field rather than individual fields, or you can use the `biblatex` options `useauthor=false`, `useeditor=false`, `usetranslator=false`, and `usecompiler=false` in the `options` field (`chaucer:alt`). If you look at the `chaucer:alt` entry in `chicago-test.bib`, you'll notice a peculiarity of this system of toggles. In order to ensure that the title of the book appears at the head of the entry, you need to use *all four* of the toggles, even though the entry contains no translator. Internally, `biblatex-chicago-notes-df` is either searching for an author-substitute, or it is skipping over elements of the ordered, unidirectional chain `author -> editor -> translator -> compiler -> title`. If you don't include `usetranslator=false` in the `options` field, then the package begins its search at `translator` and continues on to `namec`, even though you have `usecompiler=false` in `options`. The result will be that the compilers' names will appear at the head of the entry. If you want to skip over parts of the chain, you must turn off *all* of the parts up to the one you wish printed.

This system of toggles, then, can turn off `biblatex-chicago-notes-df`'s mechanism for finding a name to place at the head of an entry, but it also very usefully adds the possibility of citing a work with an author by its editor, compiler or translator instead (17.45; `eliot:pound`), something that wasn't possible before. For full details of how this works, see the `editor`type documentation below. (Of course, in `collection` and `proceedings` entry types, an author isn't expected, so there the `editor` is required, as in standard `biblatex`. Also, in `article` or `review` entries with `entrysubtype` `magazine`, the absence of an author triggers the use of the `journaltitle` in its stead. See those entry types for further details.)

**NB:** The *Manual* provides specific instructions for formatting the names of both anonymous and pseudonymous authors (17.32–39). In the former case, if no author is known or guessed at, then it may simply be omitted (`virginia:plantation`). The use of "Anonymous" as the name is "generally to be avoided," but may in some cases be useful "in a bibliography in which several anonymous works

need to be grouped.” If, on the other hand, “the authorship is known or guessed at but was omitted on the title page,” then you need to use the `author` field to let `biblatex-chicago-notes-df` know this fact. If the author is known (`horsley:prosodies`), then put `anon` in the `author` field, if guessed at (`cook:sotweed`) put `anon?` there. (In both cases, `biblatex-chicago-notes-df` tests for these *exact* strings, so check your typing if it doesn’t work.) This will have the effect of enclosing the name in square brackets, with or without the question mark indicating doubt. As long as you have the right string in the `author` field, `biblatex-chicago-notes-df` will also do the right thing automatically in the short note form.

The `nameaddon` field furnishes the means to cope with the case of pseudonymous authorship. If the author’s real name isn’t known, simply put `pseud.` (or `\bibstring{pseudonym}`) in that field (`centinel:letters`). If you wish to give a pseudonymous author’s real name, simply include it there, formatted as you wish it to appear, as the contents of this field won’t be manipulated as a name by `biblatex` (`lecarre:quest`). If you have given the author’s real name in the `author` field, then the pseudonym goes in `nameaddon`, in the form `Firstname Lastname, pseud.` (`creasey:ashe:blast, creasey:morton:hide, creasey:york:death`). This latter method will allow you to keep all references to one author’s work under different pseudonyms grouped together in the bibliography, as recommended by the *Manual*.

**author** In `biblatex-chicago-notes-df`, this field serves a function very much in keeping with the spirit of standard `biblatex`, if not with its letter. Instead of allowing you to change the string used to identify an author, the field allows you to indicate when an author is anonymous, that is, when his or her name doesn’t appear on the title page of the work you are citing. As I’ve just detailed under `author`, the *Manual* generally discourages the use of “Anonymous” as an author, preferring that you simply omit it. If, however, the name of the author is known or guessed at, then you’re supposed to enclose that name within square brackets, which is exactly what `biblatex-chicago-notes-df` does for you when you put either `anon` (author known) or `anon?` (author guessed at) in the `author` field. (Putting the square brackets in yourself doesn’t work right, hence this mechanism.) The macros test for these *exact* strings, so check your typing if you don’t see the brackets. Assuming the strings are correct, `biblatex-chicago-notes-df` will also automatically do the right thing in the short note form. Cf. `author`. (See 17.33–34; `cook:sotweed, horsley:prosodies`.)

**bookauthor** For the most part, as in `biblatex`, a `bookauthor` is the author of a booktitle, so that, for example, if one chapter in a book has different authorship from the book as a whole, you can include that fact in a reference (17.75; `will:cohere`). Keep in mind, however, that the entry type for introductions, forewords and afterwords (`customc`) uses `bookauthor` as the author of title (`polakow:afterw, prose:intro`).

**bookpagination** This, a standard `biblatex` field, allows you automatically to prefix the appropriate string to information you provide in a `pages` field. If you leave it blank, the default is to print no identifying string (the equivalent of setting it to `none`), as this is the practice the *Manual* recommends for nearly all page numbers. Even if the numbers you cite aren’t pages, but it is otherwise clear from the context what they represent, you can still leave this blank. If, however, you specifically need to identify what sort of unit the `pages` field represents, then you can either hand-format that field yourself, or use one of the provided `bibstrings` in the `bookpagination` field. These `bibstrings` currently are `column, line, paragraph, page, section, and verse`, all of which are used by `biblatex`’s standard styles.

There are two points that may need explaining here. First, all the `bibstrings` I have just listed follow the Chicago specification, which may be confusing if they don’t produce the strings you expect. Second, remember that `bookpagination`

applies only to the `pages` field — if you need to format a citation’s postnote field, then you must use `pagination`, which see (15.45–46, 17.128–138).

**booksubtitle** The subtitle for a booktitle. See the next entry for further information.

**booktitle** In the `customa`, `customb`, `inbook`, `incollection`, and `inproceedings` entry types, the `booktitle` field holds the title of the larger volume in which the title itself is contained as one part. It is important not to confuse this with the `maintitle`, which holds the more general title of multiple volumes, e.g., *Collected Works*. It is perfectly possible for one `.bib` file entry to contain all three sorts of title (`euripides:orestes`, `plato:republic:gr`). You may also find a `booktitle` in other sorts of entries (e.g., `book` or `collection`), but there it will almost invariably be providing information for the `BIBTEX` cross-referencing apparatus (`prairie:state`), which I discuss below ([crossref](#)).

**booktitleaddon** An annex to the `booktitle`. It will be printed in the main text font, without quotation marks. If your data begins with a word that would ordinarily only be capitalized at the beginning of a sentence, then simply ensure that that word is in lowercase, and `biblatex-chicago-notes-df` will automatically do the right thing.

**chapter** This field holds the chapter number, mainly useful only in an `inbook` or an `incollection` entry where you wish to cite a specific chapter of a book (`ashbrook:brain`).

**commentator** I have implemented this `biblatex` field pretty much as that package’s standard styles do, even though the *Manual* doesn’t actually mention it. It may be useful for some purposes. Cf. `annotator`.

**crossref** `Biblatex` uses the standard `BIBTEX` cross-referencing mechanism, and has also introduced a modified one of its own (`xref`). The `crossref` field works exactly the same as it always has, while `xref` attempts to remedy some of the deficiencies of the usual mechanism by ensuring that child entries will inherit no data at all from their parents. Having said all that, a few further instructions may be in order for users of both `biblatex` and `biblatex-chicago-notes-df`. First, remember that fields in a collection entry, for example, differ from those in an `incollection` entry. In order for the latter to inherit the `booktitle` field from the former, the former needs to have such a field defined, even though a collection entry has no use itself for such an entry (see `ellet:galena`, `keating:dearborn`, `lippincott:chicago`, and `prairie:state`). Note also that an entry with a `crossref` field will mechanically try to inherit all applicable fields from the entry it cross-references. In the case of `ellet:galena et al.`, you can see that this includes the subtitle field found in `prairie:state`, which would then, quite incorrectly, be added to the title of `ellet:galena`. In cases like these, you could just make sure that `prairie:state` didn’t contain such a field, by placing the entire title + subtitle in the `title` field, separated by a colon. You’d certainly need to provide a `shorttitle` field for short footnotes, if you chose this solution. Alternatively, as you can see in `ellet:galena`, you can just define an empty `subtitle` field to prevent it inheriting the unwanted subtitle from `prairie:state`.

Turning now more narrowly to `biblatex-chicago-notes-df`, the *Manual* (17.70) specifies that if you cite several contributions to the same collection, all (including the collection itself) may be listed separately in the bibliography, which the package does automatically, using the default inclusion threshold of 2 in the case both of `crossref`’ed and `xref`’ed entries. (The familiar `\nocite` command may also help in some circumstances.) In footnotes the specification suggests that, after a citation of any one contribution to the collection, all subsequent contributions may, even in the first, long footnote, be cited using a slightly shortened form, thus “avoiding clutter.” In the bibliography the abbreviated form is appropriate for all the child entries. The current version of `biblatex-chicago-notes-df` implements these instructions, but only if you use a `crossref` or

an `xref` field, and only in `incollecion`, `inproceedings`, or `customa` entries (on the last named, see just below). If you look at `ellet:galena`, `keating:dearborn`, `lippincott:chicago`, and `prairie:state` you'll see this mechanism in action in both notes and bibliography. If you wish to disable this, then simply don't use a `crossref` or `xref` field in your entries.

A published collection of letters requires a somewhat different treatment (17.78). If you cite more than one letter from the same collection, then the *Manual* specifies that only the collection itself should appear in the bibliography. In footnotes, you can use the `customa` entry type, documented above, for each individual letter, while the collection as a whole may well require a book entry. I have, after some consideration, implemented the system of shortened references in `customa` entries, even though the *Manual* doesn't explicitly require it. As with `incollecion` and `inproceedings`, mere use of a `crossref` or `xref` field will activate this mechanism, while avoidance of said fields will disable it. (See `white:ross:memo`, `white:russ`, and `white:total`, for examples of the `xref` field in action in this way, and please note that the second of these entries is entirely fictitious, provided merely for the sake of example.) How then to keep the individual letters from appearing in the bibliography? The simplest mechanism is one provided by `biblatex`, which involves the `keywords` field. Choose a keyword for any entry you wish excluded from the bibliography — I've chosen `original`, for reasons that will become clearer later — then in the optional argument to the `\printbibliography` command in your document include, e.g., `notkeyword=original`. (Cf. `keywords` and `userf`.)

If you look closely at the `.bib` entries for `white:ross:memo` and `white:russ`, you'll see that, despite using `xref` instead of `crossref`, the notes referring to them inherit data from the parent (`white:total`). The citation mechanism is making a separate call to the parent's `.bib` entry, formatting the information there to fill out the bare data provided by the child, but this only happens in `customa`, `incollecion`, and `inproceedings` entries. It is perfectly possible that other sorts of entries may make use of `crossref` or `xref` fields — `inbook` and `customb` come to mind — but such entries will not result in the activation of shortened references in notes and bibliography, nor, when using `xref`, in the inheritance I have just pointed out. This is how I interpret the specification, though I'm open to persuasion on this score.

I should also take this opportunity to mention that you need to be careful when using the shorthand field in conjunction with the `crossref` or `xref` fields, bearing in mind the complicated questions of inheritance posed by all such cross-references, most especially in `customa`, `incollecion`, and `inproceedings` entries. A shorthand field in a parent entry is, at least in the current state of `biblatex-chicago-notes-df`, a bad idea.

- date** This field may be used to specify an item's complete date of publication, in `iso8601` format, i.e., `yyyy-mm-dd`. It may also be used to specify a date range, according to Lehman's instructions in § 2.3.8 of `biblatex.pdf`. If only part of a date is required, then the month and year fields may be more convenient. The latter may be particularly useful in some entries because it can hold more than just numerical data, in contrast to `date` itself. Cf. `origdate` and `urldate`.
- day** This field, as of `biblatex 0.9`, is obsolete, and will be ignored if you use it in your `.bib` files. Use `date` instead.
- doi** Standard `biblatex` field. The Digital Object Identifier of the work, which the **Manual** suggests you can use "in place of page numbers or other locators" (17.181; `friedman:learning`). Cf. `url`.
- edition** Standard `biblatex` field. If you enter a plain cardinal number, `biblatex` will convert it to an ordinal (`chicago>manual`), followed by the appropriate string. Any other sort of edition information will be printed as is, though if your data begins

with a word (or abbreviation) that would ordinarily only be capitalized at the beginning of a sentence, then simply ensure that that word (or abbreviation) is in lowercase, and `biblatex-chicago-notes-df` will automatically do the right thing (`babb:peru`, `times:guide`). In most situations, the *Manual* generally recommends the use of abbreviations in both bibliography and notes, but there is room for the user's discretion in specific citations (`emerson:nature`).

In a previous release of `biblatex-chicago-notes-df`, I introduced the `userd` field to hold this non-numeric information, as `biblatex` only accepted an integer in the `edition` field, but this changed in version 0.8. The `userd` field is now obsolete, and will be silently ignored.

**editor** As far as possible, I have implemented this field as `biblatex`'s standard styles do, but the requirements specified by the *Manual* present certain complications that need explaining. Lehman points out in his documentation that the `editor` field will be associated with a title, a booktitle, or a maintitle, depending on the sort of entry. More specifically, `biblatex-chicago-notes-df` associates the `editor` with the most comprehensive of those titles, that is, `maintitle` if there is one, otherwise `booktitle`, otherwise `title`, if the other two are lacking. In a large number of cases, this is exactly the correct behavior (`adorno:benj`, `centinel:letters`, `plato:republic:gr`, among others). Predictably, however, there are numerous cases that require, for example, an additional editor for one part of a collection or for one volume of a multi-volume work. For these cases I have provided the `namea` field. You should format names for this field as you would for author or editor, and these names will always be associated with the title (`donne:var`).

As you will see below, I have also provided a `nameb` field, which holds the translator of a given title (`euripides:orestes`). If `namea` and `nameb` are the same, `biblatex-chicago-notes-df` will concatenate them, just as `biblatex` already does for `editor`, `translator`, and `namec` (i.e., the compiler). Furthermore, it is conceivable that a given entry will need separate editors for each of the three sorts of title. For this, and for various other tricky situations, there is the `\partedit` macro (and its siblings), designed to be used in a `note` field or in one of the `titleaddon` fields (`chaucer:liferecords`). (Because the strings identifying an editor differ in notes and bibliography, one can't simply write them out in such a field, hence the need for a macro, which I discuss further in the commands section below [5.1].) Cf. `namea`, `nameb`, `namec`, and `translator`.

**editora**  
**editorb**  
**editorc** The new release of `biblatex` provides these fields as a means to specify additional contributors to texts in a number of editorial roles. I'm uncertain how relevant they may be to users of the Chicago style, but I have implemented them just as the standard styles do. To specify the role, use the new fields `editoratype`, `editorbtype`, and `editorctype`, and see `biblatex.pdf` § 2.3.6.

**editortype** Normally, with the exception of the article and review types, `biblatex-chicago-notes-df` will automatically find a name to put at the head of an entry, starting with an author, and proceeding in order through `editor`, `translator`, and `namec` (the compiler). If all four are missing, then the title will be placed at the head. (In article and review entries with a `magazine` `entrysubtype`, a missing author immediately prompts the use of `journaltitle` at the head of an entry. See above under `article` for details.) The `editortype` field, added in `biblatex` 0.7, provides even greater flexibility, giving you the ability to put a compiler at the head of an entry without using `namec`, and even though an author is named (`eliot:pound` shows this mechanism in action for a standard editor, rather than a compiler). Two things are necessary for this to happen. First, in the `options` field you need to set `useauthor=false`, then you need to put the name you wish to see at the head of your entry into the `editor` or the `namea` field. If the "editor" is in fact a compiler, then you need to put `compiler` into the `editortype` field, and `biblatex` will print the correct string after the name in both the bibliography and in the long note form.

There are a few details of which you need to be aware. Because `biblatex-chicago-notes-df` has added the `namea` field, which gives you the ability to identify the editor specifically of a title as opposed to a maintitle or a booktitle, the `editortype` mechanism checks first to see whether a `namea` is defined. If it is, that name will be used at the head of the entry, if it isn't it will go ahead and look for an editor. When the `editor` field is used, `biblatex`'s sorting algorithms will work properly, and also its `labelname` mechanism, meaning that a shortened form of the editor will be used in the short note form. If, however, the `namea` field provides the name, then your `.bib` entry will need to have a `sortkey` field to aid in alphabetizing, and it will also need a `shorteditor` defined to help with the short note form, not a `shortauthor`, ruled out because `useauthor=false`.

In `biblatex 0.9` Lehman has reworked the string concatenation mechanism, for reasons he outlines in his `RELEASE` file, and I have followed his lead. In short, if you define the `editortype` field, then concatenation is turned off, even if the name of the editor matches, for example, that of the translator. In the absence of an `editortype`, the usual mechanisms remain in place, that is, if the `editor` exactly matches a translator and/or a `namec`, or alternatively if `namea` exactly matches a `nameb` and/or a `namec`, then `biblatex` will print the appropriate strings. The *Manual* specifically (17.41) recommends not using these identifying strings in the short note form, and `biblatex-chicago-notes-df` follows their recommendation. If you nevertheless need to provide such a string, you'll have to do it manually in the `shorteditor` field, or perhaps, in a different sort of entry, in a `shortauthor` field.

It may also be worth noting that because of certain requirements in the specification – absence of an author, for example – the `useauthor` mechanism won't work properly in the following entry types: `collection`, `customa`, `customc`, `letter`, `patent`, `periodical`, `proceedings`, `review`, `suppbook`, `suppcollection`, and `suppperiodical`.

**editortype** These fields are new to `biblatex 0.9`, and identify the exact role of the person named in the corresponding `editor[a-c]` field. Note that they are not part of the string concatenation mechanism. I have implemented them just as the standard styles do, though I am uncertain of their usefulness for users of Chicago. Cf. `biblatex.pdf` § 2.3.6.

**eid** Standard `biblatex` field, providing a string or number some journals use uniquely to identify a particular article. Only applicable to the article entry type. Not typically required by the *Manual*.

**entrysubtype** Standard and very powerful `biblatex` field, left undefined by the standard styles. In `biblatex-chicago-notes-df` it has four very specific uses, the first three of which I have designed in order to maintain, as much as possible, backward compatibility with the standard styles. First, in article, periodical, and review entries, the field allows you to differentiate between scholarly “journals,” on the one hand, and “magazines” and “newspapers” on the other. Usage is fairly simple: you need to put the exact string `magazine` into the `entrysubtype` field if you are citing one of the latter two types of source, whereas if your source is a “journal,” then you need do nothing.

The second use involves references to works from classical antiquity and, according to the *Manual*, from the Middle Ages, as well. When you cite such a work using the traditional divisions into books, sections, lines, etc., divisions which are presumed to be the same across all editions, then you need to put the exact string `classical` into the `entrysubtype` field. This has no effect in long notes or in the bibliography, but it does affect the formatting of short notes, where it suppresses some of the punctuation. Ordinarily, you will use this toggle in a book or a `customb` entry, but it is possible that a journal might well also present an edition of such a work. Given the tradition of using italics for the titles of such works, this may require using a `titleaddon` field (with hand for-

matting) instead of a title. If you wish to reference a classical or medieval work by the page numbers of a particular, non-standard edition, then you shouldn't use the `entrysubtype` toggle. Also, and the specification is quite clear about this, works from the Renaissance and later, even if cited by the traditional divisions, have short notes formatted normally, and therefore don't need an `entrysubtype` field. (See *Manual* 17.250–262; `aristotle:metaphy:gr`, `plato:republic:gr`; `euripides:orestes` is an example of a translation cited by page number in a modern edition.)

The third use occurs in `misc` entries. If such an entry contains no `entrysubtype` field, then the citation will be treated just as the standard `biblatex` styles would, including the use of italics for the title. Any string at all in `entrysubtype` tells `biblatex-chicago-notes-df` to treat the source as part of an unpublished archive. You should use the `origdate` field for specifically-dated archival material like a letter or an interview, when the `origdate` will be formatted appropriately and printed just after the title. A `misc` entry with `entrysubtype` defined is the least formatted of all those specified by the *Manual* — see above under `misc` for all the details on how these citations work.

Fourth, and finally, the field can be defined in the new `artwork` entry type in order to refer to a work from antiquity whose title you do not wish to be italicized. Please see the documentation of `artwork` above for the details.

- foreword** As with the `afterword` field above, `foreword` will in general function as it does in standard `biblatex`. Like `afterword` (and `introduction`), however, it has a special meaning in a `customc` entry, where you simply need to define it somehow (and leave `afterword` and `introduction` undefined) to make a foreword the focus of a citation.
- holder** A standard `biblatex` field for identifying a patent's holder(s), if they differ from the author. The *Manual* has nothing to say on the subject, but `biblatex-chicago-notes-df` prints it (them), in parentheses, just after the author(s).
- howpublished** Standard `biblatex` field, mainly applicable in the `booklet` entry type, where it replaces the `publisher`. I have also retained it in the `misc` and `unpublished` entry types, for historical reasons.
- institution** Standard `biblatex` field. In the `thesis` entry type, it will usually identify the university for which the thesis was written, while in a `report` entry it may identify any sort of institution issuing the report.
- introduction** As with the `afterword` and `foreword` fields above, `introduction` will in general function as it does in standard `biblatex`. Like those fields, however, it has a special meaning in a `customc` entry, where you simply need to define it somehow (and leave `afterword` and `foreword` undefined) to make an introduction the focus of a citation.
- isbn** Standard `biblatex` field, for providing the International Standard Book Number of a publication. Not typically required by the *Manual*.
- isrn** Standard `biblatex` field, for providing the International Standard Technical Report Number of a report. Only relevant to the `report` entry type, and not typically required by the *Manual*.
- issn** Standard `biblatex` field, for providing the International Standard Serial Number of a periodical in an article or a periodical entry. Not typically required by the *Manual*.
- issue** Standard `biblatex` field, designed for article, periodical, or review entries identified by something like "Spring" or "Summer" rather than by the usual month or number fields (`brown:bremer`).

**issuesubtitle** The subtitle for an `issuetitle` — see next entry.

**issuetitle** Standard `biblatex` field, intended to contain the title of a special issue of any sort of periodical. If the reference is to one article within the special issue, then this field should be used in an article entry (`conley:fifthgrade`), whereas if you are citing the entire issue as a whole, then it would go in a `periodical` entry, instead (`good:wholeissue`). The `note` field is the proper place to identify the type of issue, e.g., `special issue`, with the initial letter lower-cased to enable automatic contextual capitalization.

**journalsubtitle** The subtitle for a `journaltitle` — see next entry.

**journaltitle** Standard `biblatex` field, replacing the standard `BIBTEX` field `journal`, which, however, still works as an alias. It contains the name of any sort of periodical publication, and is found in the article and review entry types. In the case where a piece in an article or review (`entrysubtype magazine`) doesn't have an author, `biblatex-chicago-notes-df` provides for this field to be used as the author. See above (section 4.1) under **article** for details. The `lakeforester:pushcarts` and `nyt:trevorobit` entries in `chicago-test.bib` will give you some idea of how this works.

**keywords** This field is `biblatex`'s extremely powerful and flexible technique for filtering bibliography entries, allowing you to subdivide a bibliography according to just about any criteria you care to invent. See `biblatex.pdf` (3.10.4) for thorough documentation. In `biblatex-chicago-notes-df`, the field provides a convenient means to exclude certain entries from making their way into a bibliography. We have already seen (**customa**, above) how the *Manual* (17.78) requires, in the case of published collections of letters, that when more than one letter from the same collected is cited, the bibliography should contain only a reference to the collection as a whole (`white:ross:memo`, `white:russ`, `white:total`). Similarly, when citing both an original text and its translation (see **userf**, below), the *Manual* (17.66) suggests including the original at the end of the translation's bibliography entry, a procedure which requires that the original not also be printed as a separate bibliography entry (`furet:passing:eng`, `furet:passing:fr`, `aristotle:metaphy:trans`, `aristotle:metaphy:gr`). Finally, citations of well-known reference works (like the *Encyclopaedia Britannica*, for example), need only be presented in notes, and not in the bibliography (17.238–239; `ency:britannica`, `wikiped:bibtex`; see **inreference**, above). In all these cases, I have suggested the inclusion of `original` in the `keywords` field, along with a `notkeyword=original` in the optional argument to the `\printbibliography` command, though of course you can choose any key you wish.

**language** A standard `biblatex` field, designed to allow you to specify the language(s) in which a work is written. As a general rule, the Chicago style doesn't require you to provide this information, though it may well be useful for clarifying the nature of certain works, such as bilingual editions, for example. There is at least one situation, however, when the *Manual* does specify this data, and that is when the title of a work is given in translation, even though no translation of the work has been published, something that might happen when a title is in a language deemed to be unparseable by a majority of your expected readership (17.65–67, 17.166, 17.177; `pirumova`, `rozner:liberation`). In such a case, you should provide the language(s) involved using this field, connecting multiple languages using the keyword `and`. (I have retained `biblatex`'s `\bibstring` mechanism here, which means that you can use the standard `bibstrings` or, if one doesn't exist for the language you need, just give the name of the language, capitalized as it should appear in your text. You can also mix these two modes inside one entry without apparent harm.)

An alternative arrangement suggested by the *Manual* is to retain the original title of a piece but then to provide its translation, as well. If you choose this



option, you'll need to make use of the `usere` field, on which see below. In effect, you'll probably only ever need to use one of these two fields in any given entry, and in fact `biblatex-chicago-notes-df` will only print one of them if both are present, preferring `usere` over `language` for this purpose (see kern and weresz). Note also that both of these fields are universally associated with the title of a work, rather than with a `booktitle` or a `maintitle`. If you need to attach a language or a translation to either of the latter two, you could probably manage it with special formatting inside those fields themselves.

**lista** I intend this field specifically for presenting citations from reference works that are arranged alphabetically, where the name of the item rather than a page or volume number should be given. The field is a `biblatex` list, which means you should separate multiple items with the keyword `and`. Each item receives its own set of quotation marks, and the whole list will be prefixed by the appropriate string ("*s.v.*," *sub verbo*, pl. "*s.vv.*"). `Biblatex-chicago-notes-df` will only print such a field in a book or an inference entry, and you should look at the documentation of these entry types for further details. (See *Manual* 17.238–239; `ency:britannica`, `grove:sibelius`, `times:guide`, `wikiped:bibtex`.)

**location** This is `biblatex`'s version of the usual `BIBTEX` field `address`, though the latter is accepted as an alias if that simplifies the modification of older `.bib` files. According to the *Manual* (17.99), a citation usually need only provide the first city listed on any title page, though a list of cities separated by the keyword `"and"` will be formatted appropriately. If the place of publication is unknown, you can use `\autocap{n}.p.` instead (17.102), though in many or even most cases this isn't strictly necessary (17.32–34; `virginia:plantation`). For all cities, you should use the common English version of the name, if such exists (17.101).

Two more details need explanation here. In article, periodical, and review entries, there is usually no need for a `location` field, but "if a journal might be confused with another with a similar title, or if it might not be known to the users of a bibliography," then this field can present the place or institution where it is published (17.174, 17.196; `lakeforester:pushcarts`, `kimluu:diethyl`, and `garrett`). Less predictably, this is where the *Manual* indicates that a particular book is a reprint edition (17.123), so in such a case you need to use the `biblatex-chicago-notes-df` macro `\reprint`, followed by a comma, space, and the location (`aristotle:metaphy:gr`, `schweitzer:bach`). The `origdate` field may be used to give the original date of publication, and of course more complicated situations should usually be amenable to inclusion in the `note` field (`emerson:nature`).

**mainsubtitle** The subtitle for a `maintitle` — see next entry.

**maintitle** The main title for a multi-volume work, e.g., "Opera" or "Collected Works." (See `donne:var`, `euripides:orestes`, `harley:cartography`, `lach:asia`, `pelikan:christian`, and `plato:republic:gr`.)

**maintitleaddon** An annex to the `maintitle`, for which see previous entry. Such an annex would be printed in the main text font. If your data begins with a word that would ordinarily only be capitalized at the beginning of a sentence, then simply ensure that that word is in lowercase, and `biblatex-chicago-notes-df` will automatically do the right thing.

**month** Standard `biblatex` field, containing the month of publication. This should be an integer, i.e., `month={3}` not `month={March}`. See `date` for more information.

**namea** This is one of the fields `biblatex` provides for style writers to use, but which it leaves undefined itself. In `biblatex-chicago-notes-df` it contains the name(s) of the editor(s) of a title, if the entry has a `booktitle` or `maintitle`, or both, in which situation the editor would be associated with one of these latter fields (`donne:var`). You should present names in this field exactly as you would those in an au-

thor or editor field, and the package will concatenate this field with nameb if they are identical. See under **editor** above for the full details. Cf. also nameb, namec, translator, and the macros `\partedit`, `\parttrans`, `\parteditandtrans`, `\partcomp`, `\parteditandcomp`, `\parttransandcomp`, and `\partedittransandcomp`, for which see section 5.1.

**nameaddon** This field is provided by biblatex, though not used by the standard styles. In `biblatex-chicago-notes-df`, it allows you to specify that an author's name is a pseudonym, or to provide either the real name or the pseudonym itself, if the other is being provided in the author field. The abbreviation "pseud." (always lowercase in English) is specified, either on its own or after the pseudonym (`centinel:letters`, `creasey:ashe:blast`, `creasey:morton:hide`, `creasey:york:death`, and `le-carre:quest`); `\bibstring{pseudonym}` does the work for you. See under **author** above for the full details.

**nameb** Like namea, above, this is a field left undefined by the standard biblatex styles. In `biblatex-chicago-notes-df`, it contains the name(s) of the translator(s) of a title, if the entry has a booktitle or maintitle, or both, in which situation the translator would be associated with one of these latter fields (`euripides:orestes`). You should present names in this field exactly as you would those in an author or translator field, and the package will concatenate this field with namea if they are identical. See under the **translator** field below for the full details. Cf. also namea, namec, origlanguage, translator, userf and the macros `\partedit`, `\parttrans`, `\parteditandtrans`, `\partcomp`, `\parteditandcomp`, `\parttransandcomp`, and `\partedittransandcomp` in section 5.1.

**namec** The *Manual* (17.41) specifies that works without an author may be listed under an editor, translator, or compiler, assuming that one is available, and it also specifies the strings to be used with the name(s) of compiler(s). All this suggests that the *Manual* considers this to be standard information that should be made available in a bibliographic reference, so I have added that possibility to the many that biblatex already provides, such as the editor, translator, commentator, annotator, and redactor, along with writers of an introduction, foreword, or afterword. Since `biblatex.bst` doesn't offer a compiler field, I have adopted for this purpose the otherwise unused field namec. It is important to understand that, despite the analogous name, this field does not function like namea or nameb, but rather like editor or translator, and therefore if used will be associated with whichever title field these latter two would be were they present in the same entry. Identical fields among these three will be concatenated by the package, and concatenated too with the (usually) unnecessary commentator, annotator and the rest. Also please note that I've arranged the concatenation algorithms to include namec in the same test as namea and nameb, so in this particular circumstance you can, if needed, make namec analogous to these two latter, title-only fields. (See above under **editor** for details of how you may, in certain circumstances, use that field to identify a compiler. This method will be particularly useful if you don't need to concatenate the namec with any other role, because if you use the editor field biblatex will automatically attend to alphabetization and name-replacement in the bibliography.)

It might conceivably be necessary at some point to identify the compiler(s) of a title separate from the compiler(s) of a booktitle or maintitle, but for the moment I've run out of available name fields, so you'll have to fall back on the `\partcomp` macro or the related `\parteditandcomp`, `\parttransandcomp`, and `\partedittransandcomp`, on which see Commands (section 5.1) below. (Future releases may be able to remedy this.) It may be as well to mention here too that of the three names that can be substituted for the missing author at the head of an entry, `biblatex-chicago-notes-df` will choose an editor if present, then a translator if present, falling back to namec only in the absence of the other two, and assuming that the fields aren't identical, and therefore to be concatenated. In a change from the previous behavior, these algorithms also now test for namea or

nameb, which will be used instead of editor and translator, respectively, giving the package the greatest likelihood of finding a name to place at the head of an entry. Please remember, however, that if this name is supplied by any of the non-standard fields name[a-c], then you will need to provide a sortkey to assist with alphabetization in the bibliography.

**note** As in standard biblatex, this field allows you to provide bibliographic data that doesn't easily fit into any other field. In this sense, it's very like addendum, but the information provided here will be printed just before the publication data. (See `chaucer:alt`, `chaucer:liferecords`, `cook:sotweed`, `emerson:nature`, and `rodman:walk` for examples of this usage in action.) It also has a specialized use in all the periodical types (article, periodical, and review), where it holds supplemental information about a journaltitle, such as "special issue" (`conley:fifthgrade`, `good:wholeissue`). In all uses, if your data begins with a word that would ordinarily only be capitalized at the beginning of a sentence, then simply ensure that that word is in lowercase, and `biblatex-chicago-notes-df` will automatically do the right thing. Cf. `addendum`.

**number** This is a standard biblatex field, containing the number of a journaltitle in an article or review entry, the number of a title in a periodical entry, or the volume/number of a book in a series. Generally, in an article, periodical, or review entry, this will be a plain cardinal number, but in any book-like entry it may well contain considerably more information, including even a reference to "2nd ser.," for example, while the `series` field in such an entry will contain the name of the series, rather than a number. This field is also the place for the patent number in a patent entry. Cf. `issue` and `series`. (See *Manual* 17.90–95 and `boxer:china`, `palmatary:pottery`, `wauchope:ceramics`; 17.163 and `beattie:crime`, `conley:fifthgrade`, `friedman:learning`, `garrett`, `gibbard`, `hlatky:hrt`, `mcmillen:antebellum`, `rozner:liberation`, `warr:ellison`.)

**NB:** This may be an opportune place to point out that the *Manual* (17.129) prefers arabic to roman numerals in most circumstances (chapters, volumes, series numbers, etc.), even when such numbers might be roman in the work cited. The obvious exception is page numbers, in which roman numerals indicate that the citation came from the front matter, and should therefore be retained. Another possible exception is in references to works "with many and complex divisions," in which "a mixture of roman and arabic" may be "easier to disentangle."

**options** A standard biblatex field, for setting certain options on a per-entry basis rather than globally. Information about some of the more common options may be found above under `author` and below in section 6. See `chaucer:alt`, `eliot:pound`, `herwign:office`, `lecarre:quest`, and `mla:style` for examples of the field in use.

**organization** A standard biblatex field, retained mainly for use in the `misc`, `online`, and `manual` entry types, where it may be of use to specify a publishing body that might not easily fit in other categories. In `biblatex`, it is also used to identify the organization sponsoring a conference in a `proceedings` or `inproceedings` entry, and I have retained this as a possibility, though the *Manual* is silent on the matter.

**origdate** This is a new biblatex field, replacing the obsolete `origyear`, and allowing more than one full specification for those references which need to provide more than one date. As with the analogous `date` field, you provide the date (or range of dates) in ISO8601 format, i.e., `yyyy-mm-dd`. In most entry types, you would use `origdate` to provide the date of first publication of a work, most usually needed only in the case of reprint editions, but also recommended by the *Manual* for electronic editions of older works (17.123, 17.146–7; `aristotle:metaphy:gr`, `emerson:nature`, `james:ambassadors`, `schweitzer:bach`). In the `customa` and `misc` (with `entrysubtype`) entry types, the `origdate` identifies when a letter (or similar) was written, with the `date` field — for `customa` entries — identifying the date of

publication of the whole collection. If such a published collection were itself a reprint, improvisation in the location field might be able to rescue the situation. (See `jackson:paulina:letter`, `white:ross:memo`, `white:russ`, and `white:total` for how customa entries usually work; `creel:house` and `spock:interview` show the field in action in misc entries.)

Because the `origdate` field only accepts numbers, some improvisation may be needed if you wish to include “n.d.” (`\bibstring{nodate}`) in an entry. In customa and misc, this information can be placed in `titleaddon`, but in other entry types you may need to use the location field.

**origlanguage** In keeping with the *Manual*'s specifications, I have fairly thoroughly redefined biblatex's facilities for treating translations. The `origtitle` and `origlocation` fields aren't used, while the `language` and `origdate` fields have been press-ganged for other duties. The `origlanguage` field, for its part, retains a dual role in presenting translations in a bibliography. The details of the *Manual*'s suggested treatment when both a translation and an original are cited may be found below under **userf**. Here, however, I simply note that the introductory string used to connect the translation's citation with the original's is “Originally published as,” which I suggest may well be inaccurate in a great many cases, as for instance when citing a work from classical antiquity, which will most certainly not “originally” have been published in the Loeb Classical Library. Although not, strictly speaking, authorized by the *Manual*, I have provided another way to introduce the original text, using the `origlanguage` field, which must be provided *in the entry for the translation, not the original text* (`aristotle:metaphy:trans`). If you put one of the standard biblatex bibstrings there (enumerated below), then the entry will work properly across multiple languages. Otherwise, just put the name of the language there, localized as necessary, and `biblatex-chicago-notes-df` will eschew “Originally published as” in favor of, e.g., “Greek edition:” or “French edition:”. This has no effect in notes, where only the work cited — original or translation — will be printed, but it may help to make the *Manual*'s suggestions for the bibliography more palatable.

That was the first usage, in keeping at least with the spirit of the *Manual*. I have also, perhaps less in keeping with that specification, retained some of biblatex's functionality for this field. If an entry doesn't have a `userf` field, and therefore won't be combining a text and its translation in the bibliography, you can also use `origlanguage` as Lehman intended it, so that instead of saying, e.g., “translated by X,” the entry will read “translated from the German by X.” The *Manual* doesn't mention this, but it may conceivably help avoid certain ambiguities in some citations. As in biblatex, if you wish to use this functionality, you have to provide *not* the name of the language, but rather a bibliography string, which may, at the time of writing, be one of `american`, `brazilian`, `danish`, `dutch`, `english`, `french`, `german`, `greek`, `italian`, `latin`, `norwegian`, `portuguese`, `spanish`, or `swedish`, to which I've added `russian`.

**origlocation** NB: This field is now obsolete, and has, as announced previously, been replaced by `lista`, which see. Please update your `.bib` files accordingly.

**origyear** This field is, as of biblatex 0.9, obsolete. It is ignored if it appears in a `.bib` file.

**pages** This is the standard biblatex field for providing page references. In many article and review entries you'll find this contains something other than a page number, e.g. a section name or edition specification (`17.188`, `17.191`, `17.202`; `kozinn:review`, `nyt:obittrevor`, `nyt:trevorobit`). Of course, the same may be true of almost any sort of entry, though perhaps with less frequency. Curious readers may wish to look at `brown:bremer` (`17.172`) for an example of a `pages` field used to facilitate reference to a two-part journal article. Cf. `number` for more information on the *Manual*'s preferences regarding the formatting of numerals; `bookpagination` and `pagination` provide details about biblatex's mechanisms for

specifying what sort of division a given pages field contains; and users discuss a different way to present the section information pertaining to a newspaper article.

**pagination** This, a standard biblatex field, allows you automatically to prefix the appropriate identifying string to information you provide in the postnote field of a citation command, whereas bookpagination allows you to prefix a string to the pages field. Please see **bookpagination** above for all the details on this functionality, as aside from the difference just mentioned the two fields are equivalent.

**part** Standard biblatex field, which identifies physical parts of a single logical volume in book-like entries, not in periodicals. It has the same purpose in biblatex-chicago-notes-df, but because the *Manual* (17.88) calls such a thing a “book” and not a “part,” the string printed in notes and bibliography will, at least in English, be “bk.” instead of the plain dot between volume number and part number (harley:cartography, lach:asia). This field should only be used in association with a volume number, so if you need to identify “parts” or “books” that are part of a published series, for example, then you’ll need to use a different field, (which in this case would be number [palmatary:pottery]). Cf. volume.

**publisher** Standard biblatex field. Remember that “and” is a keyword for connecting multiple publishers, so if a publisher’s name contains “and,” then you should either use the ampersand (&) or enclose the whole name in additional braces. (See *Manual* 17.103–114; aristotle:metaphy:gr, cohen:schiff, creasey:ashe:blast, dunn:revolutions.)

There are, as one might expect, a couple of further subtleties involved here. Ordinarily, two publishers will be separated by a forward slash in both notes and bibliography, but if a company issues “certain books through a special publishing division or under a special imprint,” then the two names will be separated by a comma, which you will need to provide in the publisher field. The *Manual*’s example (17.112) is “Ohio University Press, Swallow Press,” which would cause biblatex-chicago-notes-df no problems. If a book has two co-publishers, “usually in different countries,” (17.113) then the simplest thing to do is to choose one, probably the nearest one geographically. If you feel it necessary to include both, then levistrauss:savage demonstrates one way of doing so, using a combination of the publisher and location fields. Finally, if the publisher is unknown, then the *Manual* recommends (17.109) simply using the place (if known) and the date. If for some reason you need to indicate the absence of a publisher, the abbreviation given by the *Manual* is n.p., though this can also stand for “no place.” Some style guides apparently suggest using s.n. (= *sine nomine*) to specify the lack of a publisher, but the *Manual* doesn’t mention this.

**redactor** I have implemented this field just as biblatex’s standard styles do, even though the *Manual* doesn’t actually mention it. It may be useful for some purposes. Cf. annotator and commentator.

**reprinttitle** **NB: Please note that this feature is in an alpha state, and that I’m contemplating using a different field in the future for this functionality. I include it here in the hope that it might receive some testing in the meantime.** At the request of Will Small, I have included a means of providing the original publication details of an essay or a chapter that you are citing from a subsequent reprint, e.g., a *Collected Essays* volume. In such a case, at least according to the *Manual* (17.73), such details needn’t be provided in notes, only in the bibliography, and then only if these details are “of particular interest.” The data would follow an introductory phrase like “originally published as,” making the problem strictly parallel to that of including details of a work in the original language alongside the details of its translation. I have addressed the latter problem with the userf field, which provides a sort of cross-referencing method for this purpose, and

reprinttitle works in *exactly* the same way. In the .bib entry for the reprint you include a cross-reference to the cite key of the original location using the reprinttitle field (which it may help mnemonically to think of as a “reprinted title” field). The main difference between the two forms is that userf prints all but the author of the original work, whereas reprinttitle suppresses both the author and the title of the original, giving only the more general details, beginning with, e.g., the journaltitle or booktitle and continuing from there. The string prefacing this information will be “Originally published in.” Please see the documentation on userf below for all the details on how to create .bib entries for presenting your data.

**series** A standard biblatex field, usually just a number in an article, periodical, or review entry, almost always the name of a publication series in book-like entries. If you need to attach further information to the series name in a book-like entry, then the number field is the place for it, whether it be a volume, a number, or even something like “2nd ser.” or “\bibstring{oldseries}.” Of course, you can also use \bibstring{oldseries} or \bibstring{newseries} in an article entry, but there you would place it in the series field itself. (In fact, the series field in article, periodical, and review entries is one of the places where biblatex allows you just to use the plain bibstring oldseries, for example, rather than making you type \bibstring{oldseries}. The type field in manual, patent, report, and thesis entries also has this auto-detection mechanism in place; see the discussion of \bibstring below for details.) In whatever entry type, these bibstrings produce the required abbreviation, which thankfully is the same in both notes and bibliography. (For books and similar entries, see *Manual* 17.90–95; boxer:china, browning:aurora, palmatary:pottery, plato:republic:gr, wauchope:ceramics; for periodicals, see 17.178; garaud:gatine, sewall:letter.) Cf. number for more information on the *Manual*’s preferences regarding the formatting of numerals.

**shortauthor** This is a standard biblatex field, but biblatex-chicago-notes-df makes considerably greater use of it than the standard styles. For the purposes of the Chicago style, the field provides the name to be used in the short form of a footnote. In the vast majority of cases, you don’t need to specify it, because the biblatex system selects the author’s last name from the author field and uses it in such a reference, but in a few cases this default behavior won’t work. In books without an author and listed under an editor, biblatex does the right thing and uses the surname of the editor in a short note (zukowsky:chicago), but if the work is listed under a compiler (or any of the non-standard names name[a-c]), you need to provide that person’s name in shortauthor, and also remember to provide a sortkey to make sure the work will be alphabetized correctly in the bibliography. (The current version of biblatex will now automatically alphabetize by translator if that is the name given at the head of an entry.) If, in an author-less article or review entry (entrysubtype magazine), you allow biblatex-chicago-notes-df to use the title of the periodical as the author — the default behavior — then your shortauthor field needs to contain the name of the periodical, formatted appropriately, which usually means something like “\mkbibemph{Periodical Title}.” Note that in this case, too, you’ll need to help the alphabetizing routines by providing a sortkey field (gourmet:052006, lakeforester:pushcarts, nyt:trevorobit).

As mentioned under editortype, the *Manual* (17.41) recommends against providing the identifying string (e.g., ed. or trans.) in the short note form, and biblatex-chicago-notes-df follows their recommendation. If you need to provide these strings in such a citation, then you’ll have to do so by hand in the shortauthor field, or in the shorteditor field, whichever you are using.

**shorteditor** Like shortauthor, a field to provide a name for a short footnote, in this case for, e.g., a collection entry that typically lacks an author. The shortauthor field works just as well in most situations, but if you have set useauthor=false (and not useeditor=false) in an entry’s options field, then only shorteditor will be recognized. Cf. editortype, above.

**shorthand** This is biblatex’s mechanism for using abbreviations in place of the usual short note form, and I’ve left it effectively unmodified in `biblatex-chicago-notes-df`, apart from a few formatting tweaks. Any entry which contains such a field will produce a normal first note, either long or short according to your package options, informing the reader that the work will hereafter be cited by this abbreviation. As in `biblatex`, the `\printshorthands` command will produce a formatted list of abbreviations for reference purposes, a list which the *Manual* suggests should be placed either in the front matter (when using footnotes) or before the endnotes, in case these are used. (See 16.39–40, and also `biblatex.pdf` for more information.)

As I mentioned above under **crossref**, extra care is needed when using shorthands with cross-references, and I would avoid them in all parent entries, at least in the current state of `biblatex-chicago-notes-df`.

**shorttitle** A standard `biblatex` field, primarily used to provide an abbreviated title for short notes. In `biblatex-chicago-notes-df`, you need to take particular care with customa entries, where, as explained above, the *Manual* requires a special format (“to Recipient”). (See 17.76–78; `jackson:paulina:letter`, `white:ross:memo`, `white:russ`.) Some `misc` entries (with an `entrysubtype`) also need special attention. (See `creel:house`, where the full title is used as the `shortauthor` + `shorttitle` by putting `\isdot` into the `shortauthor` field.) Remember, also, that the generic titles in `review` and `misc` entries may not want capitalization in all contexts, so, as with the `title` field, if you begin a `shorttitle` with a lowercase letter the style will do the right thing (`barcott:review`, `bundy:macneil`, `Clemens:letter`, `kozinn:review`, `ratliff:review`, `unsigned:ranke`).

**sortkey** A standard `biblatex` field, designed to allow you to specify how you want an entry alphabetized in a bibliography. In general, if an entry doesn’t turn up where you expect or want it, this field should provide the solution. More particularly, entries without an author or an editor, or with a corporate author beginning with the definite or indefinite article, will usually require your assistance in this way (`chaucer:alt`, `cotton:manufacture`, `gourmet:052006`, `lakeforester:pushcarts`, `nyt:obittrevor`, `nyt:trevorobit`, `silver:gawain`, `unsigned:ranke`, `virginia:plantation`). Lehman also provides **sortname**, **sorttitle**, and **sortyear** for more fine-grained control. Please consult `biblatex.pdf` for the details.

**subtitle** The subtitle for a title — see next entry.

**title** In the vast majority of cases, this field works just as it always has in `BIBTEX`, and just as it does in `biblatex`. Nearly every entry will have one, the most likely exceptions being `incollection` or `online` entries with a merely generic title, instead of a specific one (`centinel:letters`, `powell:email`). The main source of difficulties flows from the *Manual*’s rules for formatting titles, rules which also hold for `booktitles` and `maintitles`. The whole point of using a `BIBTEX`-based system is for it to do the formatting for you, and in most cases `biblatex-chicago-notes-df` does just that, surrounding titles with quotation marks, italicizing them, or occasionally just leaving them alone. When, however, a title is quoted within a title, then you need to know some of the rules. A summary here should serve to clarify them, and help you to understand when `biblatex-chicago-notes-df` might need your help in order to comply with them.

The internal rules of `biblatex-chicago-notes-df` are as follows:

**Italics:** `booktitle`, `maintitle`, and `journaltitle` in all entry types; title of artwork, book, `bookinbook`, `booklet`, `collection`, `customb`, `customc`, `inbook`, `manual`, `misc` (with no `entrysubtype`), `periodical`, `proceedings`, `report`, `suppbook`, and `suppcollection` entry types.

**Quotation Marks:** title of article, image, `incollection`, `inproceedings`, `online`, `patent`, `periodical`, `thesis`, and unpublished entry types, `issuetitle` in article, `periodical`, and `review` entry types.

**Unformatted:** `booktitleaddon`, `maintitleaddon`, and `titleaddon` in all entry types, `title of customa`, `letter`, `misc` (with an `entrysubtype`), `review`, and `suppperiodical` entry types.

Now, the rules for which entry type to use for which sort of work tend to be fairly straightforward, but in cases of doubt you can consult section 4.1 above, the examples in `chicago-test.bib`, or go to the *Manual* itself, 8.164–210. Assuming, then, that you want to present a title within a title, and you know what sort of formatting each of the two would, on its own, require, then the following rules apply:

1. Inside an italicized title, all other titles are enclosed in quotation marks and italicized, so in such cases all you need to do is provide the quotation marks using `\mkbibquote`, which will take care of any following punctuation that needs to be brought within the closing quotation mark(s) (17.58; `donne:var`, `mchugh:wake`).
2. Inside a quoted title, you should present another title as it would appear if it were on its own, so in such cases you'll need to do the formatting yourself. Within the double quotes of the title another quoted title would take single quotes — the `\mkbibquote` command does this for you automatically, and also, I repeat, takes care of any following punctuation that needs to be brought within the closing quotation mark(s). (See 17.157; `garrett`, `loften:hamlet`, `murphy:silent`, `white:callimachus`.)
3. Inside a plain title (most likely in a `review` entry or a `titleaddon` field), you should present another title as it would appear on its own, once again formatting it yourself using `\mkbibemph` or `\mkbibquote`. (`barcott:review`, `gibbard`, `osborne:poison`, `ratliff:review`, `unsigned:ranke`.)

The *Manual* provides a few more rules, as well. A word normally italicized in text should also be italicized in a quoted or plain-text title, but should be in roman (“reverse italics”) in an italicized title. A quotation used as a (whole) title (with or without a subtitle) retains its quotation marks when it is quoted or plain, but loses them when it is italicized (17.60, 17.157; `lewis`). A word or phrase in quotation marks, but that isn't a quotation, retains those marks in all title types (`kimluu:diethyl`).

Finally, please note that in all `review` (and `suppperiodical`) entries, and in `misc` entries with an `entrysubtype`, and only in those entries, `biblatex-chicago-notes-df` will automatically capitalize the first word of the title after sentence-ending punctuation, assuming that such a title begins with a lowercase letter in your `.bib` database. See `\autocap` below for more details.

**titleaddon** Standard `biblatex` intends this field for use with additions to titles that may need to be formatted differently from the titles themselves, and `biblatex-chicago-notes-df` uses it in just this way, with the additional wrinkle that it can, if needed, replace the title entirely, and this in, effectively, any entry type, providing a fairly powerful, if somewhat complicated, tool for getting `BIBTEX` to do what you want (cf. `centinel:letters`, `powell:email`). This field will always be unformatted, that is, neither italicized nor placed within quotation marks, so any formatting you may need within it you'll need to provide manually yourself. The single exception to this rule is when your data begins with a word that would ordinarily only be capitalized at the beginning of a sentence, in which case you need then simply ensure that that word is in lowercase, and `biblatex-chicago-notes-df` will automatically do the right thing. See `\autocap`, below. (Cf. `brown:bremer`, `osborne:poison`, `reaves:rosen`, and `white:ross:memo` for examples where the field starts with a lowercase letter; `morgenson:market` provides an example where the `titleaddon` field, holding the name of a regular column in a newspaper, is capitalized, a situation that is handled as you would expect.)



**translator** As far as possible, I have implemented this field as *biblatex*'s standard styles do, but the requirements specified by the *Manual* present certain complications that need explaining. Lehman points out in his documentation that the `translator` field will be associated with a title, a booktitle, or a maintitle, depending on the sort of entry. More specifically, *biblatex-chicago-notes-df* associates the translator with the most comprehensive of those titles, that is, maintitle if there is one, otherwise booktitle, otherwise title, if the other two are lacking. In a large number of cases, this is exactly the correct behavior (`adorno:benj`, `centinel:letters`, `plato:republic:gr`, among others). Predictably, however, there are numerous cases that require, for example, an additional translator for one part of a collection or for one volume of a multi-volume work. For these cases I have provided the `nameb` field. You should format names for this field as you would for author or editor, and these names will always be associated with the title (`euripides:orestes`).

I have also provided a `namea` field, which holds the editor of a given title (`euripides:orestes`). If `namea` and `nameb` are the same, *biblatex-chicago-notes-df* will concatenate them, just as *biblatex* already does for `editor`, `translator`, and `namec` (i.e., the compiler). Furthermore, it is conceivable that a given entry will need separate translators for each of the three sorts of title. For this, and for various other tricky situations, there is the `\partttrans` macro (and its siblings), designed to be used in a `note` field or in one of the `titleaddon` fields (`ratliff:review`). (Because the strings identifying a translator differ in notes and bibliography, one can't simply write them out in such a field, hence the need for a macro, which I discuss further in the commands section below [5.1].)

Finally, as I detailed above under **author**, in the absence of an author or an editor, the translator will be used at the head of an entry (`silver:gawain`), and the bibliography entry alphabetized by the translator's name, behavior that can be controlled with the `usetranslator` switch in the `options` field. Cf. `author`, `editor`, `namea`, `nameb`, and `namec`.

**type** This is a standard *biblatex* field, and in its normal usage serves to identify the type of a manual, patent, report, or thesis entry. *Biblatex* 0.7 introduced the ability, in some circumstances, to use a `bibstring` without inserting it in a `\bibstring` command, and in these entry types the `type` field works this way, allowing you simply to input, e.g., `patentus` rather than `\bibstring{patentus}`, though both will work. (See `petroff:impurity`; `herwign:office`, `murphy:silent`, and `ross:thesis` all demonstrate how the `type` field may sometimes be automatically set in such entries by using one of the standard entry-type aliases).

With the arrival of Lehman's remarkable punctuation-tracking code in *biblatex* 0.8, there can be almost no use for the `type` field as a switch for the `\custpunct` macro, so I have been able to reuse it in order to generalize the functioning of the `customc` entry type, and of its aliases `suppbook` and `suppcollection`. In such entries, you can now use the `type` field to specify what sort of supplemental material you are citing, e.g., "preface to" or "postscript to." Cf. `customc` above for the details. (See *Manual* 17.74–75; `polakow:afterw`, `prose:intro`).

You can also use the `type` field in `artwork` and `image` entries to identify the medium of the artwork or photograph, e.g., `oil on canvas` or `albumen print`. If the first word in this field would normally only be capitalized at the beginning of a sentence, then leave it in lowercase in your `.bib` file and *biblatex* will automatically do the right thing in citations. Cf. `artwork` and `image`, above, for all the details. (See `leo:madonna`, `bedford:photo`).

**url** Standard *biblatex* field, it holds the `url` of an online publication, though you can provide one for all entry types. The required  $\LaTeX$  package `url` will ensure that your documents format such references properly, in the text and in the reference apparatus.

**urldate** Standard biblatex field, it identifies exactly when you accessed a given url. This field would contain the whole date, in ISO8601 format (evanston:library, grove:sibelius, hlatky:hrt, osborne:poison, sirosh:visualcortex, wikiped:bibtex). Please note that the **urlday**, **urlmonth**, and **urlyear** fields are all now obsolete.

**usera** A supplemental biblatex field which functions in biblatex-chicago-notes-df almost as a “journaltitleaddon” field. In article, periodical, and review entries with entry-subtype magazine, the contents of this field will be placed, unformatted and between commas, after the journaltitle and before the date. The main use is for identifying the broadcast network when you cite a radio or television program (bundy:macneil), though you may also want to use it to identify the section of a newspaper in which you’ve found a particular article (morgenson:market). (See *Manual* 17.190, 17.207. As far as I can work out, newspaper section information may be placed either before the date [usera] or after it [pages]. Cp. kozinn:review [17.202] and morgenson:market [17.190]. The choice would appear to be yours.)

**userb** **NB: this field is now deprecated, mainly because it is very unlikely you will have any further need for the \custpunct macros. I leave the code, and the instructions for how to use it, in place, because it’s barely possible that a need for it might still arise.** A supplemental biblatex field, with a very specific use in biblatex-chicago-notes-df. If the occasion does arise when you need to supply some context-sensitive punctuation yourself, then usually the \custpunct command will then be needed, controlled in certain circumstances by a toggle in the type field. If, however, you already need the type field for its regular usage in a customc, manual, patent, report, or thesis entry, and if you need to control the \custpunct with a toggle, then you’ll have to use \custpunctb, toggled by putting the exact string plain in userb.

**userd** NB: this field is now obsolete. If it appears in a .bib file it will be ignored.

**usere** Another supplemental biblatex field, which biblatex-chicago-notes-df uses specifically to provide a translated title of a work, something that may be needed if you deem the original language unparseable by a significant portion of your likely readership. The *Manual* offers two alternatives in such a situation: either you can translate the title and use that translation in your title field, providing the original language in language, or you can give the original title in title and the translation in usere. If you choose the latter, you may need to provide a shorttitle so that the short note form is also parseable. Cf. **language**, above. (See 17.65–67, 17.166, 17.177; kern, weresz.)

**userf** This is the last of the supplemental fields which biblatex provides, used by biblatex-chicago-notes-df for a very specific purpose. When you cite both a translation and its original, the *Manual* (17.66) recommends that, in the bibliography at least, you combine references to both texts in one entry, though the presentation in notes is pretty much up to you. In order to follow this specification, I have provided a third cross-referencing system (the others being crossref and xref), and have chosen the name userf because it might act as a mnemonic for its function.

In order to use this system, you should start by entering both the original and its translation into your .bib file, just as you normally would. The mechanism works for any entry type, and the two entries need not be of the same type. In the entry for the *translation*, you put the cite key of the original into the userf field. In the *original’s* entry, you need to include a toggle in the keywords field that will prevent that entry from being printed separately in the bibliography — I have chosen the string original, and use notkeyword=original in the \printbibliography command, though you can use anything you want. In this standard case, the data for the translation will be printed first, followed by the string originally published as, followed by the original, author omitted,

in what amounts to the same format that the *Manual* uses for long footnotes (furet:passing:eng, furet:passing:fr). As explained above (**origlanguage**), I have also included a way to modify the string printed before the original. In the entry for the *translation*, you put the original's language in origlanguage, and instead of originally published as, you'll get French edition: or Latin edition:, etc. (aristotle:metaphy:gr, aristotle:metaphy:trans) .

- venue** Standard biblatex offers this field for use in proceedings and inproceedings entries, but I haven't yet implemented it, mainly because the *Manual* has nothing to say about it. Perhaps the organization field could be used, for the moment, instead. Anything in a venue field will be ignored.
- version** Standard biblatex field, currently only available in misc and patent entries in biblatex-chicago-notes-df.
- volume** Standard biblatex field. It holds the volume of a journaltitle in article (and some review) entries, and also the volume of a multi-volume work in many other sorts of entry. Cf. part.
- volumes** Standard biblatex field. It holds the total number of volumes of a multi-volume work, and its use in an entry triggers particular behavior in short notes referring to such an entry, which notes will not print any punctuation between the title of the work and the volume+page reference given in the optional postnote field of the relevant \cite command (17.134; meredith:letters). If this behavior is inconvenient in a particular entry, you may need to provide a shorttitle field ending in an \addcomma, though in such a case you'd need to ensure that the \cite command's postnote field contained something, as otherwise the note would end, wrongly, with a comma. (The *Manual* appears to be somewhat inconsistent on this question [cf. 16.47], so if this feature proves onerous in use I could remove it.)
- xref** A modified crossref field provided by biblatex. See **crossref**, above.
- year** Standard biblatex field. It usually identifies the year of publication, though unlike the date field it allows non-numeric input, so you can put "n.d." (or, to be language agnostic, \bibstring{nodate}) here if required, or indeed any other sort of non-numerical date information. If you can guess the date then you can include that guess in square brackets instead of, or after, the "n.d." abbreviation. Cf. bedford:photo, clark:mesopot, ross:leo, thesis:madonna.

## 5 Commands

In this section I shall attempt to document all those commands you may need when using biblatex-chicago-notes-df that I have either altered with respect to the standard provided by biblatex or that I have provided myself. Some of these, unfortunately, will make your .bib file incompatible with other biblatex styles, but I've been unable to avoid this. Any ideas for more elegant, and more compatible, solutions will be warmly welcomed.

### 5.1 Formatting Commands

These commands allow you to fine-tune the presentation of your references in both notes and bibliography. You can find many examples of their usage in chicago-test.bib, and I shall try to point you toward a few such entries in what follows. **NB:** biblatex's \mkbibquote command is now mandatory in some situations. See its entry below.

- \autocap** Version 0.8 of biblatex introduced the \autocap command, which capitalizes a word inside a note or bibliography entry if that word follows sentence-ending punctuation, and leaves it lowercase otherwise. As this command is both more

powerful and more elegant than the kludge I designed for a previous version of `biblatex-chicago-notes-df` (see `\bibstring` below), you should be aware that the use of the single-letter `\bibstring` commands in your `.bib` file is obsolete.

In order somewhat to reduce the burden on users even further, I have, following Lehman's example, implemented a new system which automatically tracks the capitalization of certain fields in your `.bib` file. I chose these fields after a non-scientific survey of entries in my own databases, so of course if you have ideas for the extension of this facility I would be most interested to hear them. In order to take advantage of this functionality, all you need do is begin the data in the appropriate field with a lowercase letter, e.g., `note = {with the assistance of X}`. If the data begins with a capital letter — and this is not infrequent — that capital will always be retained. (cf., e.g., `creel:house`, `morgenson:market`.) If, on the other hand, you for some reason need such a field always to start with a lowercase letter, then you can try using the `\isdot` macro at the start, which turns off the mechanism without printing anything itself. Here, then, is the complete list of fields where this functionality is active:

1. The **addendum** field in all entry types.
2. The **booktitleaddon** field in all entry types.
3. The **edition** field in all entry types. (Numerals work as you expect them to here.)
4. The **maintitleaddon** field in all entry types.
5. The **note** field in all entry types.
6. The **shorttitle** field in the review (suppperiodical) entry type and in the `misc` type, in the latter case, however, only when there is an `entrysubtype` defined, indicating that the work cited is from an archive.
7. The **title** field in the review (suppperiodical) entry type and in the `misc` type, in the latter case, however, only when there is an `entrysubtype` defined, indicating that the work cited is from an archive.
8. The **titleaddon** field in all entry types.
9. The **type** field in `artwork`, `customc`, `image`, `suppbook`, and `suppcollection` entry types.

In any other cases — and there are only two examples of this in `chicago-test.bib` (`centinel:letters`, `powell:email`) — you'll need to provide the `\autocap` command yourself. Indeed, if you accidentally do so in one of the above fields, it shouldn't matter at all, and you'll still get what you want, but taking advantage of the automatic provisions should at least save some typing.

**`\bibstring`** This is Lehman's very powerful mechanism to allow `biblatex` automatically to provide a localized version of a string, and to determine whether that string needs capitalization, depending on where it falls in an entry. In the first release of `biblatex-chicago-notes-df`, the style relied very heavily on this macro, particularly on an extension I provided by defining all 26 letters of the (ASCII) alphabet as `bibstrings` (`\bibstring{a}`, `\bibstring{b}`, etc.) While you should continue to use the standard, whole-word `bibstrings`, **all use of the single-letter variants I formerly provided is obsolete, and will generate an error.** This functionality has been replaced by the `\autocap` command, which does the same thing, only more elegantly. This command was designed by Philipp Lehman, and has now been included in version 0.8 of `biblatex`. For yet greater convenience I have implemented, following Lehman's example, a system automating this functionality in all of the entry fields where its use was, by my reckoning, most frequent. This means that, when you require this functionality,

all you need do is input the data in such a field starting with a lowercase letter, and `biblatex-chicago-notes-df` will do the rest with no further assistance. In my `chicago-test.bib` file, this new mechanism in effect eliminated all need for the single-letter `bibstrings` and very nearly all need for the `\autocap` command — `centinel:letters` and `powell:email` being the only exceptions. Please see `\autocap` above for full details.

I should also mention here that `biblatex` 0.7 introduced a new functionality which sometimes allows you simply to input, for example, `newseries` instead of `\bibstring{newseries}`, the package auto-detecting when a `bibstring` is involved and doing the right thing, though in all such cases either form will work. This functionality is available in the `series` field of `article`, `periodical`, and `review` entries; in the `type` field of `manual`, `patent`, `report`, and `thesis` entries; in the `location` field of `patent` entries; and in the `language` field in all entry types. These are the places, as far as I can make out, where `biblatex`'s standard styles support this feature, and I have followed suit. If Lehman generalizes it still further in a future release, I shall do the same, if possible.

**`\custpunct`** In common with other American citation styles, the *Manual* requires that the commas and periods separating units of a reference go inside any quotation marks that happen to be present. As of version 0.8c, `biblatex` contains truly remarkable code that handles this situation in very nearly complete generality, detecting punctuation after the closing quotation mark and moving it inside when necessary, and also controlling which punctuation marks can be printed after which other punctuation marks, whether quotation marks intervene or not. This functionality is now mature, and `biblatex-chicago-notes-df` relies on this code to place punctuation in the “American style,” rather than on complicated `\DeclareFieldFormat` instructions that attempt to anticipate all possible permutations. One result of this, thankfully, is that both `\custpunct` and `\custpunctb` are now basically unnecessary, as their only purpose was to supply context-appropriate punctuation inside any quotation marks that users themselves provided as part of various entry fields. A second consequence, and I’ve already recommended this in previous releases anyway, is that users now *must* use `\mkbibquote` instead of `\enquote` or the usual L<sup>A</sup>T<sub>E</sub>X mechanisms inside their `.bib` files. For further details, please see the `\mkbibquote` entry below.

I have retained the code for the `\custpunct` commands in `chicago-notes-df.cbx`, in case a particularly gnarly entry might still require them, but I have already started to re-use the `type` field, which formerly served as a switch for `\custpunct`, in other contexts (see **`artwork`**, **`customc`**, and **`image`** above).

**`\isdot`** This is a standard `biblatex` macro, but I thought I might mention it here as a convenient placeholder in entry fields that you may, for one reason or another, wish simultaneously to have defined and yet to print nothing. (See `creel:house`, `nyt:obittrevor`, `sewall:letter`, `unsigned:ranke`, and `white:total`.)

**`\letterdatelong`** I have provided this macro mainly for use in the optional postnote field of the various citation commands. When citing a letter (published or unpublished, `customa` or `misc`), it may be useful to append the date to the usual short note form in order to disambiguate references. This macro simply prints the date of a letter, or indeed of any other sort of correspondence.

**`\mkbibquote`** This is the standard `biblatex` command, which requires attention here because it is a crucial part of the mechanism of Lehman’s “American” punctuation system. If you look in `chicago-notes-df.cbx` you’ll see that the quoted fields, e.g., an article or incollection title, have this command in their formatting, which does most of the work for you. If, however, you need to provide additional quotation marks in a field — a quoted title within a title, for example — then you may need to use this command so that any following period or comma will be brought within the closing quotation marks. Its use is *required* when the quoted material comes

at the end of a field, and I recommend always using it in your .bib database, as it does no harm even when that condition is not fulfilled. A few examples from `chicago-test.bib` should help to clarify this.

In an article entry, the title contains a quoted phrase:

```
title = {Diethylstilbestrol and Media Coverage of the
        \mkbibquote{Morning After} Pill}
```

Here, because the quoted text doesn't come at the end of title, and no punctuation will ever need to be drawn within the closing quotation mark, you could instead use `\enquote{Morning After}` or even `'Morning After'`. (Note the single quotation marks here — the other two methods have the virtue of taking care of nesting for you.) All of these will produce the formatted “Diethylstilbestrol and Media Coverage of the ‘Morning After’ Pill.” Here, by contrast, is a book title:

```
title = {Annotations to \mkbibquote{Finnegans Wake}}
```

Because the quoted title within the title comes at the end of the field, and because this bibliographical unit will be separated from what follows by a period in the bibliography, then the `\mkbibquote` command is necessary to bring that period within the final quotation marks, like so: *Annotations to “Finnegans Wake.”*

Let me also add that this command interacts well with Lehman's `csquotes` package, which I highly recommend, though the latter isn't strictly necessary in texts using an American style, to which `biblatex` defaults when `csquotes` isn't loaded.

**\reprint** This and the following 7 macros all help `biblatex-chicago-notes-df` cope with the fact that many bibstrings in the Chicago system differ between notes and bibliography, the former sometimes using abbreviated forms when the latter prints them in full. In the current case, if a book is a reprint, then the macro `\reprint`, followed by a comma, should go in the location field before the city of publication (`aristotle:metaphy:gr`, `schweitzer:bach`). See **location**, above.

**NB:** The rules for employing abbreviated or full bibstrings in the *Manual* are remarkably complex, but I have attempted to make them as transparent for users as possible. In `biblatex-chicago-notes-df`, if you don't see it mentioned in this section, then in theory you should always provide an abbreviated version, using the `\bibstring` mechanism, if necessary (`babb:peru`). The standard `biblatex` bibstrings should also work (`palmatary:pottery`), and any that won't should be covered by the series of macros beginning here with `\reprint` and ending below with `\parttransandcomp`.

**\partcomp** Since the *Manual* specifies that the strings editor, translator, and compiler all require different forms in notes and bibliography, and since it mentions these three apart from all the others `biblatex` provides (`annotator`, `commentator`, et al.), and further since it may indeed happen that the available fields (`editor`, `namea`, `translator`, `nameb`, and `namec`) aren't adequate for presenting some entries, I have provided 7 macros to allow you to print the correct strings for these functions in both notes and bibliography. Their names all begin with `\part`, as originally I intended them for use when a particular name applied only to a specific title, rather than to a maintitle or booktitle (cf **name** and **nameb**, above).

In the present instance, you can use `\partcomp` to identify a compiler when `namec` won't do, e.g., in a note field or the like. In such a case, `biblatex-chicago-notes-df` will print the appropriate string in your references.

**\partedit** Use this macro when identifying an editor whose name doesn't conveniently fit into the usual fields (`editor` or `namea`). (N.B.: If you are writing in French and using `cms-french.lbx`, then currently you'll need to add either `de` or `d'` after this

command in your .bib files to make the references come out right. I'm working on this.) See `chaucer:liferecords`.

- `\partedit-andcomp` As before, but for use when an editor is also a compiler.
- `\partedit-andtrans` As before, but for when when an editor is also a translator (`ratliff:review`).
- `\partedit-transandcomp` As before, but for when an editor is also a translator and a compiler.
- `\parttrans` As before, but for use when identifying a translator whose name doesn't conveniently fit into the usual fields (`translator` and `nameb`).
- `\parttrans-andcomp` As before, but for when a translator is also a compiler.

## 5.2 Citation Commands

The `biblatex` package is particularly rich in citation commands, some of which (e.g., `\supercite(s)`, `\citeyear`) provide functionality that isn't really needed by the Chicago notes and bibliography style offered here. If you are getting unexpected behavior when using them please have a look in your .log file. A command like `\textcite`, listed in § 3.6.2 of the `biblatex` manual but not defined by `biblatex-chicago`, defaults to `\cite`, and leaves a warning in the .log. Others (e.g., `\citeauthor`), though I haven't tested them extensively, should pretty much work out of the box. What remains are the commands I have found most useful and necessary for following the *Manual's* specifications, and I document in this section any alterations I have made to these. As always, if there are standard commands that don't work for you, or new commands that would be useful, please let me know, and it should be possible to fix or add them.

A number of users have run into a problem that appears when they've used a command like `\cite` inside a `\footnote` macro, rather than using `biblatex's` `\footcite` or `\autocite`. In this situation, the automatic capitalization routines will not be in operation at the start of the footnote, so instead of "Ibid.," for example, you'll see "ibid." The solution is to use `\footcite{key}`, `\autocite{key}`, or indeed `\Cite{key}`. Cf. `\Citetitle` below, and also section 3.6 of `biblatex.pdf`.

- `\autocite` I haven't adapted this in the slightest, but I thought it worth pointing out that `biblatex-chicago-notes-df` sets this command to use `\footcite` as the default option. It is, in my experience, much the most common citation command you will use, and also works fine in its multicite form, `\autocites`.
- `\cite*` While the `\cite` command works just as you would expect it to, I have also provided a starred version for the rare situations when you might need to turn off the ibidem tracking mechanism. `Biblatex` provides very sophisticated algorithms for using "Ibid" in notes, so in general you won't find a need for this command, but in case you'd prefer a longer citation where you might automatically find "Ibid," I've provided this. Of course, you'll need to put it inside a `\footnote` command manually. (See also section 6.2, below.)
- `\Citetitle` This simply prepends `\bibsentence` to the usual `\citetitle` command. Some titles may need this for the automatic contextual capitalization facility to work correctly. (Included as standard from `biblatex 0.8d`.)
- `\citetitles` Joseph Reagle noticed that, because of the way `biblatex-chicago-notes-df` formats titles in quotation marks, using the `\citetitle` command will often get you punctuation you don't want, especially when presenting a list of titles. I've included this multicite command to enable you to present such a list, if the need arises. Remember that you'll have to put it inside a `\footnote` command manually.

- \footfullcite** Another standard biblatex command, modified to work properly with biblatex-chicago-notes-df, and provided in case you find yourself in a situation where you really need the full citation in a footnote, but where \autocite would print a short note or even “Ibid.” This may be particularly useful if you’ve chosen to use all short notes by setting the short option in the arguments to \usepackage{biblatex}, yet still feel the need for the occasional full citation.
- \fullcite** This, too, is a standard command, and it too provides a full citation, but unlike the previous command it doesn’t automatically place it in a footnote. It may be useful within long textual notes.
- \headlesscite** Matthew Lundin requested a more generalized \headlesscite macro, suppressing the author’s name in specific contexts while allowing users not to worry about whether a particular citation needs the long or short form, a responsibility thereby handed over to biblatex’s tracking mechanisms. This citation command attempts to fulfill this request. Please note that, in the short form, the result will be rather like a \citetitle command, which may or may not be what you want. Note, also, that as I have provided only the most flexible form of the command, you’ll have to wrap it in a \footnote yourself. Please see the next entry for further discussion of some of the needs this command might help address.
- \headless-fullcite** I have provided this command in case you want to print a full citation without the author’s name. The *Manual* (17.31, 17.42) suggests this for brevity’s sake in cases where that name is already obvious enough from the title, and where repetition might seem awkward (creel:house, feydeau:farces, meredith:letters, and sewall:letter). Customa (and its new alias, letter) entries — and only such entries — do this for you automatically, and of course the repetition is tolerated in bibliographies for the sake of alphabetization, but in notes this command may help achieve greater elegance, even if it isn’t strictly necessary. As I’ve provided only the most flexible form of the command, you’ll have to wrap it in a \footnote yourself.
- \shortcite** I have provided this command in case, for any reason, you specifically require the short form of a note, and biblatex thinks you want something else. Again, I’ve provided only the most flexible form of the command, so you’ll have to wrap it in a \footnote manually.

If you look at chicago-notes-df.cbx, you’ll see a number of other citation commands, but those are intended for internal use only, mainly in cross-references of various sorts. Use at your own risk.

## 6 Package Options

### 6.1 Pre-Set biblatex Options

Although a quick glance through biblatex-chicago.sty will tell you which biblatex options the package sets for you, I thought I might gather them here also for your perusal. These settings are, I believe, consistent with the specification, but you can alter them in the options to biblatex-chicago in your preamble or by loading the package via \usepackage[style=chicago-notes-df]{biblatex}, which gives you the biblatex defaults unless you redefine them yourself inside the square brackets.

- abbreviate= By default, biblatex-chicago-notes-df prints the longer bibstrings, mainly for use in the bibliography, but since notes require the shorter forms of many of them, I’ve had to define many new strings for use there.
- autocite= Biblatex-chicago-notes-df places references in footnotes by default.
- footnote
- citetracker= The citetracker for the \ifciteseen test is enabled globally.
- true



<code>alldates=comp</code>	The specification calls for the long format when presenting dates, slightly shortened when presenting date ranges.
<code>ibidtracker=constrict</code>	This enables the use of “Ibid” in notes, but only in the most strictly-defined circumstances. Whenever there might be any ambiguity, <code>biblatex</code> should default to printing a more informative reference. Remember also that you can use the <code>\cite*</code> command to disable this functionality in any given reference, or indeed one of the <code>fullcite</code> commands if you need the long note form for any reason.
<code>loccittracker=constrict</code>	This allows the package to determine whether two consecutive citations of the same source also cite the same page of that source. In such a case, <code>Ibid</code> alone will be printed, without the page reference, following the specification (16.47).
<code>pagetracker=true</code>	This enables page tracking for the <code>\iffirstonpage</code> and <code>\ifsamepage</code> commands for controlling, among other things, the printing of “Ibid.” It tracks individual pages if $\LaTeX$ is in <code>oneside</code> mode, or whole spreads in <code>twoside</code> mode.
<code>usetranslator=true</code>	This enables automatic use of the translator at the head of entries in the absence of an author or an editor. In the bibliography, the entry will be alphabetized by the translator’s surname. You can disable this functionality on a per-entry basis by setting <code>usetranslator=false</code> in the options field. Cf. <code>silver:gawain</code> .

### Pre-set `biblatex-chicago` Option

<code>usecompiler=true</code>	This option enables automatic use of the name of the compiler (in the <code>namec</code> field) at the head of an entry, usually in the absence of an author, editor, or translator, in accordance with the specification ( <i>Manual</i> 17.41). It may also, like <code>useauthor</code> , <code>useeditor</code> , and <code>usetranslator</code> , be disabled on a per-entry basis by setting <code>usecompiler=false</code> in the options field. Please remember that, because <code>namec</code> isn’t a standard <code>biblatex</code> field, this name won’t be part of its sorting algorithms, and that any entry in the bibliography headed by a <code>namec</code> will therefore need a <code>sortkey</code> or the like in order to have it appear in the correct place. (The exception to this is when you modify the editor’s identifying string using the <code>editortype</code> field, which see.)
-------------------------------	--

### Other `biblatex` Formatting Options

I’ve chosen defaults for many of the general formatting commands provided by `biblatex`, including the vertical space between bibliography items and between items in the list of shorthands (`\bibitemsep` and `\loositemsep`). I define many of these in `biblatex-chicago.sty`, and of course you may want to redefine them to your own needs and tastes. It may be as well you know that the *Manual* does state a preference for two of the formatting options I’ve implemented by default: the 3-em dash as a replacement for repeated names in the bibliography (16.103–106); and the formatting of note numbers, both in the main text and at the bottom of the page / end of the essay (superscript in the text, in-line in the notes; 16.25). The code for this last formatting is also in `biblatex-chicago.sty`, and I’ve wrapped it in a test that disables it if you are using the `memoir` class, which I believe has its own commands for defining these parameters. You can also disable it by using the `footmarkoff` package option, on which see below.

<code>maxnames=10</code> <code>minnames=7</code>	Finally, in the <code>\printbibliography</code> command at the bottom of <code>sample.tex</code> you’ll see that I’ve set these two options, which control the number of names printed in the bibliography when that number exceeds 10. These numbers follow the recommendations of the <i>Manual</i> (17.29–30), and they are different from those for use in notes. By putting these options in your <code>\printbibliography</code> command, you can use <code>biblatex</code> ’s default settings for notes, which are correct as far as the <i>Manual</i> is concerned, and then switch to these settings for the bibliography.
---	--

## 6.2 `Biblatex-chicago-notes-df` Options

These are parts of the specification that not everyone will wish to enable. All except the second can be used even if you load the package in the old way via

a call to `biblatex`, but most users can just place the appropriate string(s) in the options to the `\usepackage{biblatex-chicago}` call in your preamble.

- annotation** At the request of Emil Salim, I have added to this version of `biblatex-chicago-notes-df` the ability to produce annotated bibliographies. If you turn this option on then the contents of your `annotation` (or `annotate`) field will be printed after the bibliographical reference. (You can also use external files to store annotations – please see `biblatex.pdf` § 3.10.7 for details on how to do this.) This functionality is currently in a beta state, so before you use it please have a look at the documentation for the `annotation` field, on page 16 above.
- footmarkoff** Although the *Manual* (16.25) recommends specific formatting for footnote (and endnote) marks, i.e., superscript in the text and in-line in foot- or endnotes, Charles Schaum has brought it to my attention that not all publishers follow this practice, even when requiring Chicago style. I have retained this formatting as the default setup, but if you include the `footmarkoff` option, `biblatex-chicago-notes-df` will not alter L<sup>A</sup>T<sub>E</sub>X's (or the endnote package's) defaults in any way, leaving you free to follow the specifications of your publisher. I have placed all of this code in `biblatex-chicago.sty`, so if you load the package with a call to `biblatex` instead, then once again footnote marks will revert to the L<sup>A</sup>T<sub>E</sub>X default, but of course you also lose a fair amount of other formatting, as well. See section 7.1, below.
- natbib** This may look like the standard `biblatex` option, but to keep the coding of `biblatex-chicago.sty` simpler for the moment I have reimplemented it there, where it merely loads the compatibility file `bibnate.def`. If you load the Chicago style with `\usepackage{biblatex-chicago}`, then the option should simply read `natbib`, rather than `natbib=true`. The shorter form also works if you use `\usepackage[style=chicago-notes-df]{biblatex}`, so I hope this requirement isn't too onerous.
- noibid** At the request of an early tester, I have included this option to allow you globally to turn off the *ibidem* mechanism that `biblatex-chicago-notes-df` uses by default. Some publishers, it would appear, require this. Setting this option will mean that all possible instances of *ibid.* will be replaced by the short note form. For more fine-grained control of individual citations you'll probably want to use specialized citation commands, instead. See section 5.2.
- short** This option means that your text will only use the short note form, even in the first citation of a particular work. The *Manual* (16.3) recommends this space-saving format only when you provide a *full* bibliography, though even with such a bibliography you may feel it easier for your readers to present long first citations. If you do use the `short` option, remember that there are several citation commands which allow you to present the full reference in specific cases (see section 5.2). If your bibliography is not complete, then you should not use this option.
- shorthandibid** Chris Sparks pointed out that `biblatex-chicago-notes-df` would never use *ibid.* in the case of entries containing a `shorthand` field, but rather that consecutive references to such an entry continued to provide the shorthand, instead. The *Manual* isn't, as far as I can tell, completely clear on this question. In 17.252, discussing references to works from classical antiquity, it states that “when abbreviations are used, these rather than *ibid.* should be used in succeeding references to the same work,” but I can't make out whether this rule is specific to classical references or has more general scope. Given this ambiguity, I don't think it unreasonable to provide an option to allow printing of *ibid.* instead of the shorthand in such circumstances, though the default behavior remains the same as it always has.
- strict** This still-experimental option attempts to follow the *Manual's* recommendations (16.57) for formatting footnotes on the page, using no rule between them and the main text unless there is a run-on note, in which case a short rule intervenes to emphasize this continuation. I haven't tested this code very thoroughly, and

it's possible that frequent use of floats might interfere with it. Let me know if it causes problems.

## 7 General Usage Hints

### 7.1 Loading the Package

Acting on advice from Philipp Lehman, and also on suggestions from users of the package, I recently implemented architectural changes designed to separate basic functionality from (less universally required) formatting options. With early versions of `biblatex-chicago-notes-df`, the standard way of loading the package was via a call to `biblatex`, e.g.:

```
\usepackage[style=chicago-notes-df,strict,backend=bibtex8,%
babel=other,bibencoding=inputenc]{biblatex}
```

Now, the default way to load the package, and one that will in the vast majority of standard cases produce the same results as the old invocation, will look like this:

```
\usepackage[strict,backend=bibtex8,babel=other,%
bibencoding=inputenc]{biblatex-chicago}
```

If you read through `biblatex-chicago.sty`, you'll see that it sets a number of `biblatex` options aimed at following the Chicago specification, as well as setting a few formatting variables intended as reasonable defaults (see section 6.1, above). Some parts of this specification, however, are plainly more "suggested" than "required," and indeed many publishers, while adopting the main skeleton of the Chicago style in citations, nonetheless maintain their own house styles to which the defaults I have provided do not conform.

If you only need to change one or two parameters, this can easily be done by putting different options in the call to `biblatex-chicago` or redefining other formatting variables in the preamble, thereby overriding the package defaults. If, however, you wish more substantially to alter the output of the package, perhaps to use it as a base for constructing another style altogether, then you may want to revert to the old style of invocation above. You'll lose all the definitions in `biblatex-chicago.sty`, including those to which I've already alluded and also the code that sets the note number in-line rather than superscript in endnotes or footnotes. Also in this file is the code that calls `cms-american.lbx`, which means that you'll lose all the Chicago-specific `bibstrings` I've defined unless you provide, in your preamble, a `\DeclareLanguageMapping` command adapted for your setup, on which see section 8 below and also §§ 4.8.1 and 4.10.8 in Lehman's `biblatex.pdf`.

What you *will not* lose is the ability to call the package options `annotation`, `strict`, `short`, and `noibid` (section 6.2, above), in case these continue to be useful to you when constructing your own modifications. There's very little code, therefore, actually in `biblatex-chicago.sty`, but I hope that even this minimal separation will make the package somewhat more adaptable. Any suggestions on this score are, of course, welcome.

### 7.2 Other Hints

One useful rule, when you are having difficulty creating a `.bib` entry, is to ask yourself whether all the information you are providing is strictly necessary. The Chicago specification is a very full one, but the *Manual* is actually, in many circumstances, fairly relaxed about how much of the data from a work's title page you need to fit into a reference. Authors of introductions and afterwords, multiple publishers in different countries, the real names of authors more commonly known under pseudonyms, all of these are candidates for exclusion if you aren't making specific reference to them, and if you judge that their inclusion won't be of particular interest to your readers. Of course, any data that may be of

such interest, and especially any needed to identify and track down a reference, has to be present, but sometimes it pays to step back and reevaluate how much information you're providing. I've tried to make `biblatex-chicago-notes-df` robust enough to handle the most complex, data-rich citations, but there may be instances where you can save yourself some typing by keeping it simple.

If you are having problems with the interaction of punctuation and quotation marks in notes or bibliography, first please check that you've used `\mkbibquote` in the relevant part of your `.bib` file. If you are still getting errors, please let me know, as it may well be a bug.

For the `biblatex-chicago-notes-df` style, I have fully adopted `biblatex`'s system for providing punctuation at the end of entries. Several users noted insufficiencies in previous releases of `chicago`, sometimes related to the semicolon between multiple citations, sometimes to ineradicable periods after long notes, bugs that were byproducts of my attempt to fix other end-of-entry errors. One of the side effects of this older code was (wrongly) to put a period after a long note produced, e.g., by a command like `\footnote{\headlessfullcite}`, whereas only the "foot" cite commands (including `\autocite` in the default `chicago` set up) should do so. If you came to rely on this side effect, please note now that you'll have to put the period in yourself when explicitly calling `\footnote`, like so: `\footnote{\headlessfullcite{key}}.`

When you use abbreviations at the ends of fields in your `.bib` file (e.g., "n.d." or "Inc.,") `biblatex-chicago-notes-df` should deal automatically with adding (or suppressing) appropriate punctuation after the final dot. This includes retaining periods after such dots when a closing parenthesis intervenes, as in (n.d.). Merely entering the abbreviation without informing `biblatex` that the final dot is a dot and not a period should always work, though you do have to provide manual formatting in those rare cases when you need a comma after the author's initials in a bibliography, usually in a `misc` entry (see `house:papers`). If you find you need to provide such formatting elsewhere, please let me know.

Finally, allow me to reiterate what Philipp Lehman says in `biblatex.pdf`, to wit, use `bibtex8`, rather than standard `BT8TEX`, and avoid the cryptic errors that ensue when your `.bib` file gets to a certain size.

## 8 Internationalization

Several users have requested that, in line with analogous provisions in other "American" `biblatex` styles (e.g., `biblatex-apa` and `biblatex-mla`), I include facilities for producing a Chicago-like style in other languages. With this release I have supplied three new `lbx` files, `cms-german.lbx`, its clone `cms-ngerman.lbx`, and `cms-french.lbx`, in at least partial fulfilment of this request. This means that all of the Chicago-specific `bibstrings` are now available for documents and reference apparatuses written in French and German, with, as I intend, more languages to follow, limited mainly by my finite time and even-more-finite competence. (If you would like to provide `bibstrings` for a language in which you want to work, or indeed correct deficiencies in the `lbx` files I have prepared, please contact me.)

**babel** Using these new facilities is fairly simple. By default, and this functionality remains the same as it was in the previous release of `biblatex-chicago-notes-df`, calls to `\DeclareLanguageMapping` in `biblatex-chicago.sty` will automatically load the American strings, and also `biblatex`'s American-style punctuation tracking, when you:

1. Load `babel` with `american` as the main text language.
2. Load `babel` with `english` as the main text language.

*or*

3. Do not load `babel` at all.

(This last is a change from the biblatex defaults — cp. § 3.9.1 in biblatex.pdf — but it seems to me reasonable, in an American citation style, to expect this arrangement to work well for the majority of users.)

If, for whatever reason, you wanted to use biblatex-chicago-notes-df but retain British typographical conventions — punctuation outside of quotation marks, outer quotes single rather than double, etc. — then one possible solution at least would be to follow these three steps:

1. Load babel with the `british` option.
2. Put `\DeclareLanguageMapping{british}{cms-american}` in your preamble.
3. Edit `cms-american.lbx` so that the line reading `\InheritBibliographyExtras{american}` instead reads `\InheritBibliographyExtras{british}`.

**New!** If you want to use French or German strings in the reference apparatus, then you can:

1. Load babel with `german` or `french` as the main document language.
2. Put `\DeclareLanguageMapping{german}{cms-german}` or `\DeclareLanguageMapping{french}{cms-french}` in your document preamble.

You can also define which bibstrings to use on an entry-by-entry basis by using the `hyphenation` field in your bib file, but you will have to make sure that the Chicago-specific strings for the given language are loaded using a `\DeclareLanguageMapping` call in the preamble. Indeed, if `american` isn't the main text language when loading babel, then in order to have access to those strings you'll need `\DeclareLanguageMapping{american}{cms-american}` in your preamble, as `biblatex-chicago.sty` won't load it for you.

Three other hints may be in order here. Please note, first, that I haven't altered the standard punctuation procedures used in German or French, so commas and full stops will appear outside of quotation marks, and those quotation marks themselves will be language-specific. If, for whatever reason, you wish to follow the Chicago specification and move punctuation inside quotation marks, then you'll need a declaration of this sort in your preamble:

```
\DefineBibliographyExtras{german}{%
  \DeclareQuotePunctuation{.,}}
```

Second, depending on the nature of your bibliography database, it will only rarely be possible to process the same bib file in different languages and obtain completely satisfactory results. Fields like `note` and `addendum` will often contain language-specific information that won't be translated when you switch languages, so manual intervention will be necessary. If you suspect you may have a need to use the same bib file in different languages, you can minimize the amount of manual intervention required by using the bibstrings defined either by `biblatex` or by `biblatex-chicago-notes-df`. Here, a quick read through `chicago-test.bib` should give you an idea of what is available for this purpose — see esp. the strings `by`, `nodate`, `newseries`, `number`, `numbers`, `oldseries`, `pseudonym`, `reviewof`, `revisededition`, and `volume`, and also section 5.1 above, esp. s.v. “`\partedit`.”

Finally, the French and German bibstrings I have provided may well break with established bibliographical traditions in those languages, but my main concern when choosing them was to remain as close as possible to the quirks of the Chicago specification. If you have strong objections to any of the strings, or indeed to any of my formatting decisions, please let me know.

## 9 Interaction with Other Packages

- endnotes** For users of the `endnotes` package — or of `pagenote` — `biblatex` 0.9 offers considerably enhanced functionality. Please read Lehman’s `RELEASE` file and the documentation of the `notetype` option in `biblatex.pdf` § 3.1.2.
- memoir** Another problem I have found occurs because the `memoir` class provides its own commands for the formatting of foot- and end-note marks. By default, `biblatex-chicago-notes-df` uses superscript numbers in the text, and in-line numbers in foot- or end-notes, but I have turned this off when the `memoir` class is loaded, reasoning that users of that package may well have their own ideas about such formatting.
- ragged2e** The footnote mark code I’ve just mentioned also causes problems for the `ragged2e` package, but in this case a simple workaround is to load `biblatex` *after* you’ve loaded `ragged2e` in your document preamble.
- Xe $\LaTeX$**  Nick Andrewes alerted me to problems that appeared when he used the Xe $\LaTeX$  engine to process his files. These included spurious punctuation after quotation marks in some situations, and also failures in the automatic capitalization routines. Some of these problems disappeared when I switched to using Lehman’s punctuation-tracking code for “American” styles, but some remained. A bug report from J. P. E. Harper-Scott suggested a new way of addressing the issue, and the newest version of Lehman’s `csquotes` package (4.4) incorporates a full fix. This, thankfully, doesn’t require turning off any of Xe $\LaTeX$ ’s features, and indeed merely involves upgrading to the latest version of `csquotes`, which I recommend doing in any case. Compatibility with the EU1 encoding is now standard in that package.

## 10 Known Bugs

There are a number of things I haven’t implemented. The solution in `brown:bremer` to multi-part journal articles obviously isn’t optimal, and I should investigate a way of making it simpler. If the kludge presented there doesn’t appeal, you can always, for the time being, refer separately to the various parts. Legal citations are another thorny issue, and implementing them would involve choosing a particular documentation scheme (for which there exist at least three widely-used standards in the US), then providing what would effectively be an entirely separate `biblatex` style, bearing little or no relation to the usual look of Chicago citations. Indeed, the *Manual* (17.275) even makes it clear that you should be using a different reference book if you are presenting work in the field, so I’ve thought it prudent to stay clear of those waters. I’m open to arguments on this score, and of course if you have other issues with particular sorts of citation I’m happy to take them on board. The *Manual* covers an enormous range of materials, but if we exclude the legal citations it seems to me that the available entry types could be pressed into service to address the vast majority of them. If this optimism proves misguided, please let me know.

This release fixes the formatting errors of which I am aware, though users writing in French should be aware of problems with the `\partedit` command in section 5.1 above. There also remain the larger issues I’ve discussed throughout this documentation, which mainly represent my inability to make all of `biblatex-chicago-notes-df`’s formatting functions transparent for the user, but thankfully Lehman’s superb punctuation-tracking code has preemptively fixed a great many small errors, some of which I hadn’t even noticed before I began testing the new functionality. That there are other micro-bugs seems certain — if you report them I’ll do my best to fix them.

On my 800 MHz PIII with 256 MB of RAM (running Slackware), it has to be admitted that `biblatex-chicago-notes-df` makes  $\LaTeX$  run a bit slowly. The `bibtex8` run is swift enough, but the actual formatting of the document can lag. It may be that newer machines mostly eliminate this, or it may be that I need to pay

more attention to the efficiency of the code, but in any case I thought I should warn you in advance.

Sharp-eyed readers may also notice that many of the macro names in `chicago-notes-df.cbx` and `chicago-notes-df.bbx` hark back to an earlier version of Lehman's `biblatex`, version 0.6 to be exact. I had already written a fair amount of the code when he released version 0.7, and since I'd redefined so many of the macros I thought I might just keep the same names for the time being. If the package proves useful I'd like eventually to bring them more into line with the current state of `biblatex`, if only to ease maintenance.

## 11 Revision History

### 0.9a: Released March 20, 2010

- Quick fixes for compatibility with `biblatex` 0.9a.

### 0.9: Released March 18, 2010

Obsolete and Deprecated Features:

- The `userd` field is now obsolete. All information it used to hold should be placed in the `edition` field.
- The `origyear` field is now obsolete in `biblatex`. It has been replaced by `origdate`, and because the latter allows a full date specification, I have been able to make the operation of `customa` (= `letter`), `misc` (with an `entry-subtype`), and `patent` entries more intuitive. The `RELEASE` file contained in this package gives the short instructions on how to update your `.bib` files, and you can also consult the documentation of those entry types above.
- The modified `csquotes.cfg` file I provided in earlier releases is now obsolete, and has been removed from the package. Please upgrade to the latest version of `csquotes` and, if you are still using my modified `.cfg` file, remove it from your  $\TeX$  search path, or at the very least excise the code I provided.

Other New Features:

- Added the files `cms-german.lbx` (with its clone `cms-ngerma.lbx`) and `cms-french.lbx`, which allow the creation of Chicago-like references in those languages. See section 8 above for details on usage.
- Added the `annotation` package option to allow the creation of annotated bibliographies. This code is still not entirely polished yet, but it is usable. Please see page 16 above for instructions and hints.
- Added `biblatex`'s new `bookinbook` entry type, which currently functions as an alias of the `customb` type. As `biblatex` now provides standard equivalents for all of the custom types I initially found it necessary to provide — `letter` = `customa`, `bookinbook` = `customb`, and `suppbook` & `suppcollection` = `customc` — it may soon be time to prune out the custom types to enhance compatibility with other `biblatex` styles. I shall give plenty of warning before I do so.
- In line with the new system adopted in `biblatex` 0.9, using the `editortype` field turns off the usual string concatenation mechanisms of the Chicago style. See Lehman's `RELEASE` file for a discussion of this.
- I have added support for the new `editor[a-c]` and `editor[a-c]type` fields, and they work just as in standard `biblatex`, though I'm uncertain how much use they'll get from users of the Chicago style.

- I have added many bibstrings to the .ltx files to help with internationalization. The new ones that you might want to use in your .bib files include: pseudonym, nodate, revisededition, numbers, and reviewof. Please see section 8 for a fuller list.

#### **0.8.9d: Released February 17, 2010**

- Chris Sparks and Aaron Lambert both found formatting bugs in the 0.8.9c code. I've fixed these bugs, and am releasing this version now, the last in the 0.8.9 series. The next release of biblatex-chicago-notes-df, due as soon as possible, will contain many more significant changes, including those necessary for it to function properly with the recently-released biblatex version 0.9. In the meantime, at least version 0.8.9d should produce more accurate output.

#### **0.8.9c: Released November 4, 2009**

- Emil Salim noticed that the *ibidem* mechanism wasn't working properly, printing the page number after "Ibid" even when the page reference of the preceding citation was identical. The fix for this involved setting loccittracker=constrict in biblatex-chicago.sty, something you'll have to do manually yourself if you're loading the package via a call to biblatex rather than to biblatex-chicago.
- Several users have reported unwanted behavior when repeated names in bibliographies are replaced with the bibnamedash. This release should fix both when the bibnamedash appears and what punctuation follows it.

#### **0.8.9b: Released September 9, 2009**

- Fixed a long-standing bug in formatting names in the bibliography. The package now correctly places a comma after the reversed name that begins the entry, using biblatex's \revsdnamedelim command. Many thanks to Johanna Pink for catching my rather egregious error.
- While fixing some formatting errors that cropped up when using the newest version of biblatex (0.8h at time of writing), I also spotted some more venerable bugs in the code for using shortened cross-references for citing multiple entries in a collection of essays or letters. I believe this now works correctly, but please let me know if you discover differently.
- Joseph Reagle noticed that endnote marks (produced using the endnotes package) did not receive the same treatment as footnote marks. I have rectified this, placing the code in biblatex-chicago.sty so that you can turn it off either by using the old package-loading system or by setting the footmarkoff package option when loading biblatex-chicago.
- Updates to Lehman's csquotes package have rendered my modifications in csquotes.cfg obsolete. Please use the latest version of csquotes (4.4a at time of writing) and ignore my file, which will disappear in a later release.
- At the request of Will Small, I have included some code, still in an alpha state, to allow you to specify, in the bibliography, the original publication details of essays which you are citing from later reprints (a *Collected Essays* volume, for example). See the documentation above under the reprinttitle field if you would like to test this functionality.

#### **0.8.9a: Released July 5, 2009**

- Slight changes for compatibility with biblatex 0.8e. The package still works with 0.8c and 0.8d, as well.



## 0.8.9: Released July 2, 2009

### Obsolete and Deprecated Features:

- The **single-letter bibstrings** (`\bibstring{a}`, `\bibstring{b}`, etc.) are now obsolete. You should replace any still present in your .bib file with `\autocap` commands — see § 3.8.4 of `biblatex.pdf`.

### Other New Features:

- The default way of loading the package is now with `\usepackage[further-options]{biblatex-chicago}` rather than `\usepackage[style=chicago-notes-df,further-options]{biblatex}`. Please see section 7.1 above for details and hints.
- Package-specific bibstrings have been removed from the .cbx and .bbx files and are now gathered in a new file, **cms-american.lbx**, which changes the way the package interacts with **babel**. It is now somewhat simpler if you want the defaults, but somewhat more complex if you require non-standard features. Please see section 9 above for more details.
- Two new entry types have been added: **artwork** for works of visual art excluding photographs, and **image** for photographs. See the documentation of artwork for how to create .bib entries for both types.
- Added the new bibliography and entry option **usecompiler**, set to true by default. This streamlines the code that finds a name to head an entry (**author** -> **editor** [**or namea**] -> **translator** [**or nameb**] -> **compiler** [**namec**] -> **title**). The whole system should work more consistently now, but do see the author and namec documentation for improved notes on how to use it.
- Added the new bibliography option **footmarkoff**, to turn off the optional in-line (as opposed to superscript) formatting of the marks in foot- or endnotes. You only need this if you load the package with the new default `\usepackage{biblatex-chicago}`; users loading it the old way get default L<sup>A</sup>T<sub>E</sub>X formatting.
- At Matthew Lundin's request, I have added the citation command `\headlesscite`, which works like `\headlessfullcite` but allows biblatex to decide whether to print the full or the short note.
- Fully adopted biblatex's system for providing end-of-entry punctuation, which should solve some of the bugs users have been finding. See section 7.2, above, and do please let me know if inconsistencies remain.
- Added a modified **csquotes.cfg** file to address issues users were having when using the XeL<sup>A</sup>T<sub>E</sub>X engine in combination with biblatex-chicago. See section 9, above.
- Added `natbib` option to allow users of the default setup to continue to benefit from biblatex's natbib compatibility code. Thanks to Bennett Helm for pointing out this issue.
- Added a **shorthandibid** option to allow the printing of *ibid.* in consecutive references to an entry that contains a shorthand field. Thanks to Chris Sparks for calling my attention to this problem.
- While investigating the preceding, I noticed failures when combining the short option with a shorthand field. The package now actually does what it has always claimed to do under **shorthand**.
- Many small bug fixes and improvements to the documentation.

To Do:

- The shorthand vs *ibid.* question may need more careful addressing in some cross references, and also in relation to the `noibid` package option.
- Charles Schaum has quite rightly pointed out the inconsistency in my naming conventions — `biblatex-chicago.sty` as opposed to `chicago-notes-df.cbx`, for example. I'm going to delay a decision on which way to go with this until a later release.

#### 0.8.5a: Released June 14, 2009

- Quick and dirty fixes to bibliography strings to allow compatibility with `biblatex` version 0.8d. If you are still using 0.8c, then I would wait for the next version of `biblatex-chicago-notes-df`, which is due soon. See README.

#### 0.8.5: Released January 10, 2009

Obsolete and Deprecated Features:

- The `\custpunct` commands are now deprecated — Lehman's "American" punctuation tracking facilities should handle quoted text automatically, assuming you remember always to use `\mkbibquote` in your database. If you still need `\custpunct`, please let me know, as it may be an error in the style.
- With `\custpunct` no longer needed, the toggles activated by placing "plain" in the `type` or `userb` fields are also deprecated.

Other New Features:

- At least `biblatex 0.8b` is now required — 0.8c works fine, as well.
- I now *strongly recommend* that you use `babel` with "american" as the main text language. See section 9 above for further details.
- The `customc` entry type has been revised, allowing you to cite any sort of supplementary material using the `type` field instead of relying on toggles in the introduction, afterword, and foreword fields, though these latter still work. The two new entry types `suppbook` and `suppcollection` are both aliased to `customc`, and therefore work in exactly the same way.
- The new entry type `suppperiodical` is aliased to `review`.
- The new entry type `letter` is aliased to `customa`.
- In `inference` entries the `postnote` field of all `\cite` commands is now treated like data in `lista`, that is, it will be placed within quotation marks and prefaced with the appropriate string. The only difference is that you can only put one such article name in `postnote`, as it isn't a list field.
- I've set the new `biblatex` option `usetranslator` to `true` by default, which means entries will automatically be alphabetized by their translator in the absence of an author or an editor.
- A host of small formatting errors were eliminated, nearly all of them through adopting Lehman's punctuation tracker.
- In the main body of this documentation, I've added some **color coding** to help you more quickly to identify entry types and fields that are either new or that have undergone significant revision.

To Do:

- Separate out "options" from the basic citation "style," using a  $\LaTeX$  style file. This is an architectural change recommended by Lehman.

### 0.8.2.2: Released November 24, 2008

- Fixed spurious commas appearing in some bibliography entries, spotted by Nick Andrewes. While investigating this I noticed a more general problem with punctuation after italicized titles ending with question marks or exclamation points. This will be addressed in forthcoming revisions both of biblatex and of this package.
- Nick also reported some problems with spurious punctuation in the bibliography when using XeLaTeX. I haven't yet been able to pin down the exact cause of these, but if you are using XeLaTeX and are having (or have solved) similar problems I'd be interested to hear from you.

### 0.8.2: Released November 3, 2008

- Fixed several formatting glitches between citations in multicite commands (spotted by Joseph Reagle) and also after some prenotes.

### 0.8.1: Released October 22, 2008

Obsolete and Deprecated Features:

- The **origlocation** field is now obsolete, and has been replaced by **lista**. Please update your .bib files accordingly.
- The single-letter **\bibstring** commands I provided in version 0.7 are now deprecated. In most cases, you'll be able to take advantage of the automatic contextual capitalization facilities introduced in this release, but if you still need the single-letter **\bibstring** functionality then you should switch to **\autocap**, as I shall be removing the single-letter **\bibstrings** in a future release. See above under **\autocap** for all the details.
- The **userd** field is now deprecated, as biblatex 0.8 allows all forms of data to be included in the **edition** field. I shall be removing **userd** in a future release, so please update your .bib files as soon as is convenient.

Other New Features:

- Updated the .bbx and .cbx files to work with biblatex 0.8. This most recent version of biblatex is now required for biblatex-chicago-notes-df to work.
- Added the **usera** field, which holds supplemental information about a journaltitle in article and review entries. See the documentation of the field for details.
- Added the **\citetitles** multicite command to fix a problem with spurious punctuation when multiple titles were listed.
- Added the **\Citetitle** command to help with automatic capitalization of titles when they occur at the beginning of a note.
- Minor punctuation fixes in biblatex-chicago-notes-df.bbx.

To Do:

- Integrate biblatex's American punctuation facilities.
- Separate out "options" from the basic citation "style," using a L<sup>A</sup>T<sub>E</sub>X style file. This is an architectural change recommended by Lehman.
- Investigate and possibly integrate the new entry types provided in biblatex 0.8.

### 0.7: First public release, September 18, 2008