

bath-bst: Harvard referencing style as recommended by the University of Bath Library

Maintainer: Alex Ball*

Package v2.0 – 23 April 2018

1 Introduction

This package provides a **BibTeX** style to format reference lists in the **Harvard style** recommended by the University of Bath Library. It should be used in conjunction with **natbib** for citations.

1.1 Installation

You can use this style simply by copying the `bath.bst/bathx.bst` files into your working directory, that is, the directory holding the main `.tex` file for your document. If you want the style to be available for all your documents without having to copy it over each time, you can install it using the instructions below.

1.1.1 Automated way

A makefile is provided which you can use with the Make utility on UNIX-like systems:

- Running `make source` generates the derived files
 - `README.md`
 - `bath.bst` and `bathx.bst`
 - `bath-bst-v1.tex`
 - `bath-bst.bib` and `bath-bst-v1.bib`
 - `bath-bst.ins`
- Running `make` generates the above files and also `bath-bst.pdf` and `bath-bst-v1.pdf`.
- Running `make inst` installs the files in the user's TeX tree. You can undo this with `make uninst`.
- Running `make install` installs the files in the local TeX tree. You can undo this with `make uninstall`.

1.1.2 Manual way

You only need to follow the first two steps if you have made your own changes to the `.dtx` file. The compiled files you need are included in the zip archive.

1. Run `luatex bath-bst.dtx` to generate the source files.
2. Compile `bath-bst.dtx` with **LuaLaTeX** and BibTeX to generate the documentation. You will need, among other things, the `luatexja`, `adobemapping` and `ipaex` packages installed; this is just for the documentation, not for the BibTeX style itself. To generate the version 1 tests and documentation, compile `bath-bst-v1.tex` with LuaLaTeX and BibTeX.

*To contact the maintainer about this package, please visit the repository where the code is hosted: <https://github.com/alex-ball/bathbib>.

3. If you are using TeX Live, find your home TeX tree using the following command at the command prompt/terminal:

```
kpsewhich -var-value=TEXMFHOME
```

If you are using MikTeX, consult the MikTeX manual entry for [integrating local additions](#). You can use one of the roots (TeX trees) already defined – preferably one of the User roots – or set up a new one.

4. Move the files to your TeX tree as follows:
 - source/bibtex/bath-bst: bath-bst.dtx, (bath-bst.ins)
 - bibtex/bst/bath-bst: bath.bst, bathx.bst
 - doc/bibtex/bath-bst: bath-bst.pdf, bath-bst-v1.pdf, README.md
5. You may then have to update your installation’s file name database before TeX and friends can see the files.

2 Choosing which variant to use

The data model offered by the standard Bib_T_EX styles, and even the extended natbib variants, is not really rich enough to support the nuances of the Harvard (Bath) style. This means design decisions have to be made about whether to attempt some level of compatibility with other styles or craft something utterly unique.

In the first version of bath-bst, the intention was to set things up so that, if the same .bib file was used with a different style, the features peculiar to the Harvard (Bath) style would be ignored and the remaining information would come out in a sensible arrangement. The recommendations of the style’s documentation were to use the standard entry types as much as possible (though more semantic aliases were provided), and minimal new fields were introduced. For the more exotic (in Bib_T_EX terms) demands of the style, extensive use was made of the (standard) note and (non-standard) titleaddon fields to place information properly. If you have written a .bib file according to the principles in that first version, the bath style will still work as advertised for you.

The second version introduces a new variant, bathx (‘Bath extended’), which has a different aim. The idea with this one is that .bib files written for this style will be rendered just the same by the companion biblatex-bath style. Where possible, features from the latter have been emulated for Bib_T_EX; otherwise, the ‘cheats’ used in this style will also work under biblatex.

Happily, it has been possible to do this while keeping most of the code in common. The main differences between two versions are as follows:

- In bath, the titleaddon field is printed bare, while in bathx it is wrapped in square brackets.
- In bathx online items are marked with ‘[Online]’ automatically, while in bath you have to mark them thus yourself.
- In bathx undated items are marked with ‘n.d.’ automatically, while in bath you have to give ‘n.d.’ as the value of year yourself.

One other change worth noting regards URL access dates. In version 1, the advice was to put them in urldate. With version 2 the advice is now to put them, perhaps counter-intuitively, in urlyear. The reason is that biblatex will complain if the value of urldate is in the wrong format, but will happily print the contents of urlyear literally. If you don’t care about biblatex compatibility, you can continue to use urldate.

3 Using the style

To use the style, add these lines to your preamble:

```
\usepackage{natbib}
\newcommand*{\urlprefix}{Available from: }
\newcommand*{\urldateprefix}{Accessed }
\bibliographystyle{bathx}
```

Remember also to specify your .bib file at the end of the document:

```
\bibliography{file}
```



To make a citation in the text, use the key that corresponds to the entry in your .bib file:

```
While collections can be supplemented by other means \citep{williams1997edd},
the absence of an invisible collection amongst historians is noted by
\citet[p.556]{stieg1981inh}. It may be, as \citet{burchard1965hhl} points out,
that they have no assistants or are reluctant to delegate, or it may be
down to economic factors \citep{adams2009tc1, adams2014tc2, gb.pa2014,
adams2017tc3}\dots
```

While collections can be supplemented by other means (Williams, 1997), the absence of an invisible collection amongst historians is noted by Stieg (1981b, p.556). It may be, as Burchard (1965) points out, that they have no assistants or are reluctant to delegate, or it may be down to economic factors (Adams, 2009, 2014; *Pensions Act*, 2014; Adams, 2017)...

Please refer to the documentation for `natbib` for the full range of commands available for in-text citations. Be aware that the `natbib` option `sort` will sort citations in bibliography order, rather than the chronological order demanded by Harvard (Bath) – so don't use it!



4 Examples

The examples below are shown in three parts. The first, marked with , shows an extract from the *Harvard (Bath) guide to citing and referencing* or *Referencing images*. The second, marked with , shows the reference as formatted by BibTeX. The last shows how the reference was entered in the .bib file. The bottom right corner shows the source of the example: 'HBS' indicates the 'Specific examples' section of the Guide (followed by the number of the example); 'ref. list' indicates the 'How to organise a reference list' section of the Guide; 'RI' indicates *Referencing images*.

Some examples are highlighted in orange. This indicates that some fields have been 'abused' to achieve the right effect; in other words, they contain information that does not conform with their intended use. Some others make use of the `note` and `titleaddon` fields to achieve the right effect, where other styles might need the information placed differently. Particular care should be taken with such items when switching between different styles, though of course any item might need adjustment to take account of differing conventions.

4.1 Books and book chapters

Book with author(s)

-  Rang, H.P., Dale, M.M., Ritter, J.M., Flower, R.J. and Henderson, G., 2012. *Rang and Dale's pharmacology*. 7th ed. Edinburgh: Elsevier Churchill Livingstone.
-  Rang, H.P., Dale, M.M., Ritter, J.M., Flower, R.J. and Henderson, G., 2012. *Rang and Dale's pharmacology*. 7th ed. Edinburgh: Elsevier Churchill Livingstone.

```
@book{rang.etal2012rdp,
  author = {Rang, H. P. and Dale, M. M. and Ritter, J. M. and Flower, R. J. and Henderson, G.},
  year = {2012},
  title = {Rang and {Dale's} Pharmacology},
  edition = {7},
  address = {Edinburgh},
  publisher = {Elsevier Churchill Livingstone}}
```

HBS 1

- 📖 Open University, 1972. *Electricity and magnetism*. Bletchley: Open University Press.
- ⚙️ Open University, 1972. *Electricity and magnetism*. Bletchley: Open University Press.

```
@book{ou1972em,
  author = {Open University},
  year = {1972},
  title = {Electricity and Magnetism},
  address = {Bletchley},
  publisher = {Open University Press}}
```

HBS 1

Book with editor(s) instead of author(s)

- 📖 Rothman, K.J., Greenland, S. and Lash, T.L., eds., 2008. *Modern epidemiology*. 3rd ed. Philadelphia, Pa.: Lippincott Williams & Wilkins.
- ⚙️ Rothman, K.J., Greenland, S. and Lash, T.L., eds., 2008. *Modern epidemiology*. 3rd ed. Philadelphia, Pa.: Lippincott Williams & Wilkins.

```
@book{rothman.etal2008me,
  editor = {Kenneth J. Rothman and Sander Greenland and Timothy L. Lash},
  year = {2008},
  title = {Modern Epidemiology},
  edition = {3},
  address = {Philadelphia, Pa.},
  publisher = {Lippincott Williams & Wilkins}}
```

HBS 4

Book usually known by title rather than author

- 📖 *Oxford English dictionary*, 1989. 2nd ed. Oxford: Clarendon Press.
- ⚙️ *Oxford English dictionary*, 1989. 2nd ed. Oxford: Clarendon Press.

```
@book{oed1989,
  title = {Oxford {English} Dictionary},
  year = {1989},
  edition = {2},
  address = {Oxford},
  publisher = {Clarendon Press}}
```


HBS 5


- 📖 *The Merck index online*, n.d. [Online]. London: RSC Publishing. Available from: <http://www.rsc.org/Merck-Index> [Accessed 16 June 2016].
- ⚙️ *The Merck index online*, n.d. [Online]. London: RSC Publishing. Available from: <http://www.rsc.org/Merck-Index> [Accessed 16 June 2016].

```
@book{merckNDidx,
  title = {The {Merck} Index Online},
  address = {London},
  publisher = {RSC Publishing},
  url = {http://www.rsc.org/Merck-Index},
  urlyear = {16 June 2016}}
```

HBS 5

Electronic book


 Haynes, W.M., ed., 2014. *CRC handbook of chemistry and physics* [Online]. 94th ed. Boca Raton, Fla.: CRC Press/Taylor and Francis. Available from: <http://www.hbcnetbase.com> [Accessed 16 June 2016].

 Haynes, W.M., ed., 2014. *CRC handbook of chemistry and physics* [Online]. 94th ed. Boca Raton, Fla.: CRC Press/Taylor and Francis. Available from: <http://www.hbcnetbase.com> [Accessed 16 June 2016].

```
@book{haynes2014crc,  
  editor = {Haynes, W. M.},  
  year = {2014},  
  title = {{CRC} Handbook of Chemistry and Physics},  
  edition = {94},  
  address = {Boca Raton, Fla.},  
  publisher = {CRC Press/Taylor and Francis},  
  url = {http://www.hbcnetbase.com},  
  urlyear = {16 June 2016}}
```

HBS 3


 Hodds, J., 2016. *Referencing ebooks* [Kindle version 4.18]. Bath: University of Bath.


 Hodds, J., 2016. *Referencing ebooks* [Kindle version 4.18]. Bath: University of Bath.

```
@book{hodds2016re,  
  author = {Hodds, J.},  
  year = {2016},  
  title = {Referencing ebooks},  
  titleaddon = {Kindle version 4.18},  
  address = {Bath},  
  publisher = {University of Bath}}
```

HBS 3

One chapter/paper from a collection (by different authors) in an edited book

 Burchard, J.E., 1965. How humanists use a library. In: C.F.J. Overhage and J.R. Harman, eds. *Intrex: report on a planning conference and information transfer experiments*. Cambridge, Mass.: MIT Press, pp.41–87.

 Burchard, J.E., 1965. How humanists use a library. In: C.F.J. Overhage and J.R. Harman, eds. *Intrex: report on a planning conference and information transfer experiments*. Cambridge, Mass.: MIT Press, pp.41–87.

```
@incollection{burchard1965hhl,  
  author = {Burchard, J. E.},  
  year = {1965},  
  title = {How Humanists use a Library},  
  editor = {C. F. J. Overhage and J. R. Harman},  
  booktitle = {Intrex: report on a planning conference and information transfer experiments},  
  address = {Cambridge, Mass.},  
  publisher = {MIT Press},  
  pages = {41-87}}
```

ref. list

- 📖 Reid, D.R., 1967. Physical testing of polymer films. In: S.H. Pinner, ed. *Modern packaging films*. London: Butterworths, pp.143–183.
- ⚙️ Reid, D.R., 1967. Physical testing of polymer films. In: S.H. Pinner, ed. *Modern packaging films*. London: Butterworths, pp.143–183.

```
@incollection{reid1967ptp,
  author = {D. R. Reid},
  year = {1967},
  title = {Physical Testing of Polymer Films},
  editor = {S. H. Pinner},
  booktitle = {Modern Packaging Films},
  address = {London},
  publisher = {Butterworths},
  pages = {143-183}}
```

HBS 2

4.2 Articles and periodicals

Journal article

- 📖 Stieg, M.F., 1981a. Continuing education and the reference librarian in the academic and research library. *Library Journal*, 105(22), pp.2547–2551.
- ⚙️ Stieg, M.F., 1981a. Continuing education and the reference librarian in the academic and research library. *Library Journal*, 105(22), pp.2547–2551.

```
@article{stieg1981cer,
  author = {Stieg, M. F.},
  year = {1981},
  title = {Continuing Education and the Reference Librarian in the Academic and Research Library},
  journal = {Library Journal},
  volume = {105},
  number = {22},
  pages = {2547-2551}}
```

[ref. list](#)

- 📖 Stieg, M.F., 1981b. The information needs of historians. *College and research libraries*, 42(6), pp.549–560.
- ⚙️ Stieg, M.F., 1981b. The information needs of historians. *College and Research Libraries*, 42(6), pp.549–560.

```
@article{stieg1981inh,
  author = {Stieg, M. F.},
  year = {1981},
  title = {The Information Needs of Historians},
  journal = {College and Research Libraries},
  volume = {42},
  number = {6},
  pages = {549-560}}
```

[ref. list](#)


- 📖 Newman, R., 2010. Malaria control beyond 2010. *Brit. Med. J.*, 341(7765), pp.157–208.
- ⚙️ Newman, R., 2010. Malaria control beyond 2010. *Brit. Med. J.*, 341(7765), pp.157–208.

```
@article{newman2010mcb,
  author = {Newman, R.},
  year = {2010},
  title = {Malaria control beyond 2010},
  journal = {Brit.\@ Med.\@ J.},
  volume = {341},
  number = {7765},
  pages = {157-208}}
```

HBS 6

Electronic journal article


 Williams, F., 1997. Electronic document delivery: a trial in an academic library. *Ariadne* [Online], 10. Available from: <http://www.ariadne.ac.uk/issue10/edd/> [Accessed 5 December 1997].


 Williams, F., 1997. Electronic document delivery: a trial in an academic library. *Ariadne* [Online], 10. Available from: <http://www.ariadne.ac.uk/issue10/edd/> [Accessed 5 December 1997].

```
@article{williams1997edd,  
  author = {Williams, F.},  
  year = {1997},  
  title = {Electronic Document Delivery: a Trial in an Academic Library},  
  journal = {Ariadne},  
  volume = {10},  
  url = {http://www.ariadne.ac.uk/issue10/edd/},  
  urlyear = {5 December 1997}}
```

HBS 7


Preprint in a digital repository

 Shah, I. and Corrick, I. 2016. *How should central banks respond to non-neutral inflation expectations?* Bath: University of Bath. *OPUS* [Online]. Available from: <http://opus.bath.ac.uk> [Accessed 4 May 2016].

 Shah, I. and Corrick, I., 2016. *How should central banks respond to non-neutral inflation expectations?* Bath: University of Bath. *OPUS* [Online]. Available from: <http://opus.bath.ac.uk> [Accessed 4 May 2016].

```
@techreport{shah.corrick2016hsc,  
  author = {Shah, I. and Corrick, I.},  
  year = {2016},  
  title = {How should central banks respond to non-neutral inflation expectations?},  
  address = {Bath},  
  institution = {University of Bath},  
  library = {OPUS},  
  url = {http://opus.bath.ac.uk},  
  urlyear = {4 May 2016}}
```


HBS 8

 The Harvard (Bath) style treats preprints as grey literature, and so the `techreport` entry type is the best match semantically.

Newspaper article

 Give the issue's day and month as the volume.

 Haurant, S., 2004. Britain's borrowing hits £1 trillion. *The Guardian*, 29 July, p.16c.

 Haurant, S., 2004. Britain's borrowing hits £1 trillion. *The Guardian*, 29 July, p.16c.

```
@article{haurant2004bbh,  
  author = {Haurant, S.},  
  year = {2004},  
  title = {Britain's Borrowing Hits \pounds 1 Trillion},  
  journal = {The Guardian},  
  volume = {29 July},  
  pages = {16c}}
```

HBS 12

- 📖 The Independent, 1992. Picking up the bills. *The Independent*, 4 June, p.28a.
- ⚙️ The Independent, 1992. Picking up the bills. *The Independent*, 4 June, p.28a.

```
@article{independent1992pub,
  author = {{The Independent}},
  year = {1992},
  title = {Picking Up the Bills},
  journal = {The Independent},
  volume = {4 June},
  pages = {28a}}
```

HBS 12

4.3 Conference papers

- ❗ As in the standard BibTEX styles, `conference` is a legacy (and highly deprecated) alias for `inproceedings`.

Conference paper (when proceedings have a named editor)

- 📖 Crawford, G.I., 1965. Oxygen in metals. In: J.M.A. Lenihan and S.J. Thompson, eds. *Activation analysis: proceedings of a NATO Advanced Study Institute, 2–4 August 1964* Glasgow. London: Academic Press, pp.113–118.
- ⚙️ Crawford, G.I., 1965. Oxygen in metals. In: J.M.A. Lenihan and S.J. Thompson, eds. *Activation analysis: Proceedings of a NATO Advanced Study Institute, 2–4 August 1964* Glasgow. London: Academic Press, pp.113–118.

```
@inproceedings{crawford1965oim,
  author = {Crawford, G. I.},
  year = {1965},
  title = {Oxygen in Metals},
  editor = {J. M. A. Lenihan and S. J. Thompson},
  booktitle = {Activation Analysis: Proceedings of a {NATO} {Advanced} {Study} {Institute}},
  eventyear = {2--4 August 1964},
  venue = {Glasgow},
  address = {London},
  publisher = {Academic Press},
  pages = {113-118}}
```

HBS 10

Conference paper (when proceedings have no named editor or are part of a major series)


- 📖 Soper, D., 1972. Review of bracken control experiments with asulam. *Proceedings of the 11th British Weed Control Conference*, 15–17 November 1972 Brighton. Brighton: University of Sussex, pp.24–31.
- ⚙️ Soper, D., 1972. Review of bracken control experiments with asulam. *Proceedings of the 11th British Weed Control Conference*, 15–17 November 1972 Brighton. Brighton: University of Sussex, pp.24–31.


```
@inproceedings{soper1972rbc,
  author = {Soper, D.},
  year = {1972},
  title = {Review of Bracken Control Experiments with Asulam},
  booktitle = {Proceedings of the 11th {British} {Weed} {Control} {Conference}},
  eventyear = {15--17 November 1972},
  venue = {Brighton},
  address = {Brighton},
  publisher = {University of Sussex},
  pages = {24-31}}
```

HBS 11

4.4 Grey literature


Thesis/dissertation

 Burrell, J.G., 1973. *The importance of school tours in education*. Thesis (M.A.). Queen's University, Belfast.


 Burrell, J.G., 1973. *The importance of school tours in education*. Thesis (M.A.). Queen's University, Belfast.


```
@mastersthesis{burrell1973ist,
  author = {Burrell, J. G.},
  year = {1973},
  title = {The Importance of School Tours in Education},
  type = {Thesis (M.A.)},
  school = {Queen's University, Belfast}}
```

HBS 13

 Internally, `thesis` is an alias for `phdthesis`.


Report

 UNESCO, 1993. *General information programme and UNISIST*. Paris: UNESCO, (PGI-93/WS/22).


 UNESCO, 1993. *General information programme and UNISIST*. Paris: UNESCO, (PGI-93/WS/22).

```
@techreport{unesco1993gip,
  author = {{UNESCO}},
  year = {1993},
  title = {General Information Programme and {UNISIST}},
  address = {Paris},
  institution = {UNESCO},
  number = {PGI-93/WS/22}}
```


HBS 14

 Internally, `report` is an alias for `techreport`.

Standard


 Unless it already has one, give the standard identifier as the `author`. This is not terribly semantic but is the easiest way to get the bibliography sorting and citation text correct.

 BS 5605:1990. *Recommendations for citing and referencing published material*. BSI.


 BS 5605:1990. *Recommendations for citing and referencing published material*. BSI.

```
@standard{bs5605:1990,
  author = {{BS 5605:1990}},
  sortyear = {1990},
  title = {Recommendations for citing and referencing published material},
  organization = {BSI}}
```

HBS 15

 To suppress the printing of 'n.d.', you can either give a value for `sortyear` (the value has no other effect in this style) or you can use the `nonodate` option. Please bear in mind that the `options` field from `biblatex` has been emulated rather crudely, so you can only give one option at a time:

```
options = {nonodate},
```

 Internally, `standard` is an alias for `manual`.

Patent

- 📄 Phillipp Morris Inc., 1981. *Optical perforating apparatus and system*. European patent application 0021165A1. 1981-01-07.
- ⚙️ Phillipp Morris Inc., 1981. *Optical perforating apparatus and system*. European patent application 0021165A1. 1981-01-07.

```
@patent{pm1981opa,  
  author = {{Phillipp Morris Inc.}},  
  year = {1981},  
  title = {Optical perforating apparatus and system},  
  type = {European patent application},  
  number = {0021165A1. 1981-01-07}}
```

HBS 16

- 🔍 Internally, `patent` is an alias for `manual`.

4.5 Audiovisual materials

Image

- 📄 NASA, 2015, *NASA Astronaut Tim Kopra on Dec. 21 Spacewalk*. Washington: NASA. Available from: <http://www.nasa.gov/image-feature/nasa-astronaut-tim-kopra-on-dec-21-spacewalk> [Accessed 7 January 2015].
- ⚙️ NASA, 2015. *NASA astronaut Tim Kopra on Dec. 21 spacewalk* [Online]. Washington: NASA. Available from: <http://www.nasa.gov/image-feature/nasa-astronaut-tim-kopra-on-dec-21-spacewalk> [Accessed 7 January 2015].

```
@image{nasa2015nat,  
  author = {{NASA}},  
  year = {2015},  
  title = {{NASA} Astronaut {Tim} {Kopra} on {Dec.\@} 21 Spacewalk},  
  address = {Washington},  
  publisher = {NASA},  
  url = {http://www.nasa.gov/image-feature/nasa-astronaut-tim-kopra-on-dec-21-spacewalk},  
  urlyear = {7 January 2015}}
```

RI

- 💡 You can use either the `publisher` or the `organization` field to record the source of the image. If you use both, `organization` will be ignored.

- 📄 Iliff, D., 2006. *Royal Crescent in Bath, England - July 2006*, San Francisco: Wikimedia Foundation. Available from: https://commons.wikimedia.org/wiki/File:Royal_Crescent_in_Bath,_England_-_July_2006.jpg [Accessed 7 January 2016].
- ⚙️ Iliff, D., 2006. *Royal Crescent in Bath, England - July 2006* [Online]. San Francisco: Wikimedia Foundation. Available from: https://commons.wikimedia.org/wiki/File:Royal_Crescent_in_Bath,_England_-_July_2006.jpg [Accessed 7 January 2016].

```
@image{iliff2006rcb,  
  author = {D. Iliff},  
  year = {2006},  
  title = {{Royal} {Crescent} in {Bath,} {England} - {July} 2006},  
  address = {San Francisco},  
  organization = {Wikimedia Foundation},  
  url = {https://commons.wikimedia.org/wiki/File:Royal_Crescent_in_Bath,_England_-_July_2006.jpg},  
  urlyear = {7 January 2016}}
```

RI

📖 Anon., 1946. *Prototype electric bicycle displayed at the ‘Britain Can Make It’ exhibition organised by the Council of Industrial Design and held at the Victoria and Albert Museum, London, 1946. Designed by B. G. Bowden* [photograph]. At: London. Design Council Slide Collection.

⚙️ Anon., 1946. *Prototype electric bicycle displayed at the ‘Britain Can Make It’ exhibition organised by the Council of Industrial Design and held at the Victoria and Albert Museum, London, 1946. Designed by B. G. Bowden* [photograph]. At: London. Design Council Slide Collection.

```
@image{anon1946peb,  
  author = {Anon.},  
  year = {1946},  
  title = {Prototype electric bicycle displayed at the \enquote{Britain Can Make It} exhibition  
    organised by the {Council} of {Industrial} {Design} and held at the {Victoria} and {Albert}  
    {Museum}, {London}, 1946. {Designed} by {B.-G.} {Bowden}},  
  titleaddon = {photograph},  
  address = {London},  
  library = {Design Council Slide Collection}}
```

RI

💡 Use the `library` field to record the archive and register number. If you also wish to provide the name of the organisation that published the image, use the `publisher` field if the location you provide relates to the publisher, but the `organization` field if the location relates to the archive.

Map

📖 Andrews, J. and Dury, A., 1773. *Map of Wiltshire*, 1 inch to 2 miles. Devizes: Wiltshire Record Society.

⚙️ Andrews, J. and Dury, A., 1773. *Map of Wiltshire*. 1 inch to 2 miles. Devizes: Wiltshire Record Society.

```
@manual{andrews.dury1773wilts,  
  author = {Andrews, J. and Dury, A.},  
  year = {1773},  
  title = {Map of {Wiltshire}},  
  series = {1 inch to 2 miles},  
  address = {Devizes},  
  publisher = {Wiltshire Record Society}}
```

HBS 18

💡 The book entry type would also work for this reference.

Film, video or DVD

📖 *Macbeth*, 1948. Film. Directed by Orson Welles. USA: Republic Pictures.

⚙️ *Macbeth*, 1948. Film. Directed by Orson Welles. USA: Republic Pictures.

```
@video{macbeth1948,  
  year = {1948},  
  title = {Macbeth},  
  type = {Film},  
  note = {Directed by Orson Welles},  
  address = {USA},  
  publisher = {Republic Pictures}}
```

HBS 19

📘 Internally, `movie` and `video` are aliases for `booklet`.


Streamed video (YouTube, TED Talks, etc.)


 Moran, C., 2016. *Save Our Libraries* [Online]. Available from: <https://www.youtube.com/watch?v=gKTfCz4JtVE&feature=youtu.be> [Accessed 29 April 2016]

 Moran, C., 2016. *Save Our Libraries* [Online]. Available from: <https://www.youtube.com/watch?v=gKTfCz4JtVE&feature=youtu.be> [Accessed 29 April 2016].

```
@video{moran2016sol,  
  author = {Moran, C.},  
  year = {2016},  
  title = {Save {Our} {Libraries}},  
  url = {https://www.youtube.com/watch?v=gKTfCz4JtVE&feature=youtu.be},  
  urlyear = {29 April 2016}}
```

HBS 20


 University of Bath, 2015. *What is engineering?* Available from: <https://www.youtube.com/watch?v=NoyZarq-Zbo> [Accessed 12 January 2016].


 University of Bath, 2015. *What is engineering?* [Online]. Available from: <https://www.youtube.com/watch?v=NoyZarq-Zbo> [Accessed 12 January 2016].

```
@video{uob2015wie,  
  author = {{University of Bath}},  
  year = {2015},  
  title = {What is Engineering?},  
  url = {https://www.youtube.com/watch?v=NoyZarq-Zbo},  
  urlyear = {12 January 2016}}
```

RI

Television or radio broadcast

 *Rick Stein's French Odyssey: Episode 5*, 2006. TV. BBC2, 23 August. 20.30 hrs.

 *Rick Stein's French Odyssey: Episode 5*, 2006. TV. BBC2, 23 August. 20.30 hrs.

```
@video{rsfo2006ep5,  
  year = {2006},  
  title = {Rick {Stein's} {French} {Odyssey}: Episode 5},  
  type = {TV},  
  note = {BBC2, 23 August. 20.30 hrs}  
}
```


HBS 21


 *The Archers*, 2006. Radio. BBC Radio 4, 23 August. 19.02 hrs.

 *The Archers*, 2006. Radio. BBC Radio 4, 23 August. 19.02 hrs.

```
@audio{archers20060823,  
  year = {2006},  
  title = {The {Archers}},  
  type = {Radio},  
  note = {BBC Radio 4, 23 August. 19.02 hrs}}
```

HBS 21

 Use `type` for the medium and `note` for the channel, date and time.

 Internally, `audio` and `music` are aliases for `booklet`.

Music score

- 📖 Beethoven, L. van, 1950. *Symphony no.1 in C, Op.21*. Harmondsworth: Penguin.
- ⚙️ Beethoven, L. van, 1950. *Symphony no.1 in C, Op.21*. Harmondsworth: Penguin.

```
@book{beethoven1950symp1,  
  author = {Ludwig van Beethoven},  
  year = {1950},  
  title = {Symphony no.1 in {C,} {Op.21}},  
  address = {Harmondsworth},  
  publisher = {Penguin}}
```

HBS 22

4.6 Digital media

Website/webpage

While it is normal when using BibTeX to use misc for websites, with this style you should use online (or electronic or www) instead. Internally these are all aliases for manual.

- 📖 Holland, M., 2002. *Guide to citing internet sources* [Online]. Poole: Bournemouth University. Available from: http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html [Accessed 4 November 2002].
- ⚙️ Holland, M., 2002. *Guide to citing internet sources* [Online]. Poole: Bournemouth University. Available from: http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html [Accessed 4 November 2002].

```
@online{holland2002gci,  
  author = {Holland, M.},  
  year = {2002},  
  title = {Guide to Citing Internet Sources},  
  address = {Poole},  
  organization = {Bournemouth University},  
  url = {http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html},  
  urlyear = {4 November 2002}}
```

HBS 9

- 📖 Wiltshire Council, 2015. *Get Wiltshire walking* [Online]. Trowbridge: Wiltshire Council. Available from: <http://www.wiltshire.gov.uk/leisureandrecreation/sportphysicalactivity/getwiltshirewalking.html> [Accessed 19 August 2015].
- ⚙️ Wiltshire Council, 2015. *Get Wiltshire walking* [Online]. Trowbridge: Wiltshire Council. Available from: <http://www.wiltshire.gov.uk/leisureandrecreation/sportphysicalactivity/getwiltshirewalking.html> [Accessed 19 August 2015].

```
@online{wiltshire2015gww,  
  author = {{Wiltshire Council}},  
  year = {2015},  
  title = {Get {Wiltshire} walking},  
  address = {Trowbridge},  
  organization = {Wiltshire Council},  
  url = {http://www.wiltshire.gov.uk/leisureandrecreation/sportphysicalactivity/getwiltshirewalking.html},  
  urlyear = {19 August 2015}}
```

HBS 9

Email discussion lists (jiscmail/listserv etc.)

📄 Clark, T., 5 July 2004. A European UK Libraries Plus? *Lis-link* [Online]. Available from: lis-link@jiscmail.ac.uk [Accessed 30 July 2004].

⚙️ Clark, T., 5 July 2004. A European UK Libraries Plus? *Lis-link* [Online]. Available from: lis-link@jiscmail.ac.uk [Accessed 30 July 2004].

```
@letter{clark2004euk,  
  author = {Clark, T.},  
  year = {5 July 2004},  
  title = {A {European} {UK} {Libraries} {Plus}?},  
  journal = {Lis-link},  
  url = {lis-link@jiscmail.ac.uk},  
  urlyear = {30 July 2004}}
```

HBS 23

💡 Use the `journal` field to specify the mailing list. This triggers the change in formatting with respect to the following example. Internally the example above is transformed to an `article` entry, while the one below is transformed into an `unpublished` entry.

🔧 You will need to put the full date in the `year` field; unfortunately this means you have to put in extra work to show only the year in citations:

```
\citetext{\citeauthor[2004]{clark2004euk}}
```

(Clark, 2004)

Personal emails

📄 Alston, S., 19 July 2004. *Society of Architectural Historians of GB*. Email to K.M. Jordan.

⚙️ Alston, S., 19 July 2004. *Society of Architectural Historians of GB*. Email to K.M. Jordan.

```
@letter{alston2004sah,  
  author = {Alston, S.},  
  year = {19 July 2004},  
  title = {Society of {Architectural} {Historians} of {GB}},  
  howpublished = {Email to K.M. Jordan}}
```

HBS 24

Database

📄 Bureau van Dijk, 2008. *BT Group plc company report. FAME* [Online]. London: Bureau van Dijk. Available from: <http://www.portal.euromonitor.com> [Accessed 6 November 2014].

⚙️ Bureau van Dijk, 2008. *BT Group plc company report. FAME* [Online]. London: Bureau van Dijk. Available from: <http://www.portal.euromonitor.com> [Accessed 6 November 2014].

```
@online{bvd2008bt,  
  author = {{Bureau van Dijk}},  
  year = {2008},  
  title = {{BT} {Group} PLC Company Report},  
  library = {FAME},  
  address = {London},  
  organization = {Bureau van Dijk},  
  url = {http://www.portal.euromonitor.com},  
  urlyear = {6 November 2014}}
```

HBS 40

Dataset

- 📖 Wilson, D., 2013. *Real geometry and connectedness via triangular description: CAD example bank* [Online]. Bath: University of Bath. Available from: <https://doi.org/10.15125/BATH-00069> [Accessed 20 April 2016].
- ⚙️ Wilson, D., 2013. *Real geometry and connectedness via triangular description: CAD example bank*. Bath: University of Bath. Available from: <https://doi.org/10.15125/BATH-00069> [Accessed 20 April 2016].

```
@dataset{wilson2013rgc,  
  author = {Wilson, D.},  
  year = {2013},  
  title = {Real Geometry and Connectedness via Triangular Description: {CAD} Example Bank},  
  address = {Bath},  
  organization = {University of Bath},  
  doi = {10.15125/BATH-00069},  
  urlyear = {20 April 2016}}
```

HBS 41

- 📖 Internally, `dataset` is an alias for `manual`.

Computer program

- 📖 @screencasto, n.d. *Screencast-O-Matic* (v.2) [computer program]. Available from: <https://screencast-o-matic.com/> [Accessed 16 May 2016].
- ⚙️ @screencasto, n.d. *Screencast-O-Matic* (v.2) [computer program]. Available from: <https://screencast-o-matic.com/> [Accessed 16 May 2016].

```
@software{screencasto,  
  author = {@screencasto},  
  title = {{Screencast-O-Matic}},  
  version = {2},  
  titleaddon = {computer program},  
  url = {https://screencast-o-matic.com/},  
  urlyear = {16 May 2016}}
```

HBS 42

- 📖 Internally, `software` is an alias for `manual`.

4.7 Works in languages other than English

Work in translation

- 📖 Aristotle, 2007. *Nicomachean ethics* (W.D. Ross. Trans.). South Dakota: NuVisions.
- ⚙️ Aristotle, 2007. *Nicomachean ethics* (W.D. Ross. Trans.). South Dakota: NuVisions.

```
@book{aristotle2007ne,  
  author = {Aristotle},  
  year = {2007},  
  title = {Nicomachean Ethics},  
  translator = {W. D. Ross},  
  address = {South Dakota},  
  publisher = {NuVisions}}
```

HBS 26

Work in the Roman alphabet

📖 Esquivel, L., 2003. *Como agua para chocolate* [Like water for chocolate]. Barcelona: Debolsillo.

⚙️ Esquivel, L., 2003. *Como agua para chocolate* [Like water for chocolate]. Barcelona: Debolsillo.

```
@book{esquivel2003cap,  
  author = {Esquivel, L.},  
  year = {2003},  
  title = {Como Agua para Chocolate},  
  titleaddon = {Like water for chocolate},  
  address = {Barcelona},  
  publisher = {Debolsillo}}
```

HBS 27

📖 Thurfjell, W., 1975. Vart har vårän doktor tagit vägen? [Where has our doctor gone?] *Läkartidningen* 72, p.789.

⚙️ Thurfjell, W., 1975. Vart har vårän doktor tagit vägen? [Where has our doctor gone?]. *Läkartidningen*, 72, p.789.

```
@article{thurfjell1975vhv,  
  author = {Thurfjell, W.},  
  year = {1975},  
  title = {Vart har vårän doktor tagit vägen?},  
  titleaddon = {Where has our doctor gone?},  
  journal = {Läkartidningen},  
  volume = {72},  
  pages = {789}}
```

HBS 27

Work in a non-Roman alphabet

- 💡 The following example tricks BibTeX into treating the original rendering of the author's name as the 'von' part of a Roman-alphabet name. This requires the use of a command that simply gobbles its argument, which you have to define yourself:

```
\newcommand*{\noop}[1]{}
```

For the trick to work, the argument you give to `\noop` must be lowercase, but otherwise it can be anything you like.

📖 Hua, L. 華林甫, 1999. Qingdai yilai Sanxia diqu shuihan zaihai de chubu yanjiu 清代以來三峽地區水旱災害的初步研 [A preliminary study of floods and droughts in the Three Gorges region since the Qing dynasty], *Zhongguo shehui kexue* 中國社會科學, 1, pp.168–79.

⚙️ Hua, L. 華林甫, 1999. Qingdai yilai Sanxia diqu shuihan zaihai de chubu yanjiu 清代以來三峽地區水旱災害的初步研 [A preliminary study of floods and droughts in the Three Gorges region since the Qing dynasty]. *Zhongguo shehui kexue* 中國社會科學, 1, pp.168–79.

```
@article{hua1999qys1,  
  author = {Linfu \noop{h}華林甫 Hua},  
  year = {1999},  
  title = {Qingdai yilai {Sanxia} diqu shuihan zaihai de chubu yanjiu  
    {清代以來三峽地區水旱災害的初步研}},  
  titleaddon = {A preliminary study of floods and droughts in the {Three} {Gorges} region since  
    the {Qing} dynasty},  
  journal = {Zhongguo shehui kexue \textup{中國社會科學}},  
  volume = {1},  
  pages = {168-79}}
```


💡 If the name is due to appear initial first (e.g. after ‘In:’), you can append the non-Roman characters to the author’s surname; to do this, use inverted name order as you would for English double-barrelled names without hyphens, e.g. Hua 華林甫, Linfu.

📖 Hua, L., 1999. Qingdai yilai Sanxia diqu shuihan zaihai de chubu yanjiu [A preliminary study of floods and droughts in the Three Gorges region since the Qing dynasty], *Zhongguo shehui kexue*, 1, pp.168–79.

⚙️ Hua, L., 1999. Qingdai yilai Sanxia diqu shuihan zaihai de chubu yanjiu [A preliminary study of floods and droughts in the Three Gorges region since the Qing dynasty]. *Zhongguo shehui kexue*, 1, pp.168–79.

```
@article{hua1999qys2,
  author = {Hua, Linfu},
  year = {1999},
  title = {Qingdai yilai {Sanxia} diqu shuihan zaihai de chubu yanjiu},
  titleaddon = {A preliminary study of floods and droughts in the {Three} {Gorges} region since
    the {Qing} dynasty},
  journal = {Zhongguo shehui kexue},
  volume = {1},
  pages = {168-79}}
```

4.8 Legal references: UK legislation and parliamentary reports

Act of Parliament (UK Statutes) before 1963

📖 *Witchcraft Act 1735* (9 Geo.2, c.5).

⚙️ *Witchcraft Act 1735* (9 Geo.2, c.5).

```
@legislation{gb.wa1735,
  title = {Witchcraft {Act}},
  year = {1735},
  series = {9 Geo.2},
  chapter = {5}}
```

HBS 32

Act of Parliament (UK Statutes) 1963 onwards



📖 *Pensions Act 2014*, c.19. London: TSO.

⚙️ *Pensions Act 2014*, c.19. London: TSO.

```
@legislation{gb.pa2014,
  title = {Pensions {Act}},
  year = {2014},
  chapter = {19},
  address = {London},
  publisher = {TSO}}
```

HBS 33


House of Commons/House of Lords bill

-  Great Britain. Parliament. House of Commons, 1988. *Local government finance bill*. London: HMSO (Bills | 1987/88, 66).
-  Great Britain. Parliament. House of Commons, 1988. *Local government finance bill*. London: HMSO (Bills | 1987/88, 66).

```
@legislation{gb.bill1987/88-66,  
  author = {{Great Britain. Parliament. House of Commons}},  
  year = {1988},  
  title = {Local Government Finance Bill},  
  address = {London},  
  publisher = {HMSO},  
  series = {{Bills | 1987/88}},  
  number = {66}}
```

HBS 31

Statutory instrument

-  *The Human Medicines Regulations 2012* [Online], No.1916, United Kingdom: HMSO. Available from: http://www.legislation.gov.uk/uksi/2012/1916/pdfs/uksi_20121916_en.pdf [Accessed 17 April 2016].
-  *The Human Medicines Regulations 2012* [Online], No.1916, United Kingdom: HMSO. Available from: http://www.legislation.gov.uk/uksi/2012/1916/pdfs/uksi_20121916_en.pdf [Accessed 17 April 2016].



```
@legislation{gb.hmr2012,  
  entrysubtype = {secondary},  
  title = {The {Human} {Medicines} {Regulations}},  
  year = {2012},  
  number = {No.1916},  
  address = {United Kingdom},  
  publisher = {HMSO},  
  url = {http://www.legislation.gov.uk/uksi/2012/1916/pdfs/uksi_20121916_en.pdf},  
  urlyear = {17 April 2016}}
```

HBS 35

💡 Use the entrysubtype 'secondary' to put the number in the right place.

House of Commons paper

Use this form for reports of House of Commons select committees.

-  Great Britain. Parliament. House of Commons, 2004. *National Savings investment deposits: account 2002–2003*. London: National Audit Office (HC 2003/04, 30).
-  Great Britain. Parliament. House of Commons, 2004. *National Savings investment deposits: account 2002–2003*. London: National Audit Office (HC 2003/04, 30).

```
@techreport{gb.hc2003/04-30,  
  author = {{Great Britain. Parliament. House of Commons}},  
  year = {2004},  
  title = {National {Savings} Investment Deposits: account 2002--2003},  
  address = {London},  
  publisher = {National Audit Office},  
  series = {HC 2003/04},  
  number = {30}}
```

HBS 29

House of Lords paper

Use this form for reports of House of Lords select committees.

- 📖 Great Britain. Parliament. House of Lords, 1987. *Social fund (Maternity and Funeral Expenses) Bill*. London: HMSO (HL 1986/87, (66)).
- ⚙️ Great Britain. Parliament. House of Lords, 1987. *Social fund (Maternity and Funeral Expenses) Bill*. London: HMSO (HL 1986/87, (66)).

```
@techreport{gb.hl1986/87-66,
  author = {{Great Britain. Parliament. House of Lords}},
  year = {1987},
  title = {Social Fund ({Maternity} and {Funeral} {Expenses}) {Bill}},
  address = {London},
  publisher = {HMSO},
  series = {HL 1986/87},
  number = {66}}
```

HBS 30

- 🔧 For joint committees, you will have to hack this slightly, putting the session years in `series` and the HL and HC numbers in `number`.

Command paper

- 📖 Great Britain. Ministry of Defence, 2004. *Delivering security in a changing world: defence white paper*. London: TSO (Cm. 6041).
- ⚙️ Great Britain. Ministry of Defence, 2004. *Delivering security in a changing world: defence white paper*. London: TSO (Cm. 6041).

```
@techreport{gb.cm6041,
  author = {{Great Britain. Ministry of Defence}},
  year = {2004},
  title = {Delivering Security in a Changing World: defence White Paper},
  address = {London},
  publisher = {TSO},
  series = {Cm},
  number = {6041}}
```

HBS 34

4.9 Legal references: EU legislation and reports

EU regulation or directive, decision, recommendation or opinion

- 📖 Directive (EU) 2015/413 of the European Parliament and of the Council of 11th March 2015 facilitating cross-border exchange of information on road-safety-related traffic offences [2015] *OJ* L68/9.
- ⚙️ Directive (EU) 2015/413 of the European Parliament and of the Council of 11th March 2015 facilitating cross-border exchange of information on road-safety-related traffic offences [2015] *OJ* L68/9.

```
@legislation{eu.dir2015/413,
  title = {Directive ({EU}) 2015/413 of the {European} {Parliament} and of the {Council} of
    11th {March} 2015 Facilitating Cross-Border Exchange of Information on Road-Safety-Related
    Traffic Offences},
  year = {2015},
  journal = {OJ},
  series = {L},
  volume = {68},
  pages = {9}}
```

HBS 38

- 🔧 Use `\defcitealias` to provide a suitable citation string:

```
\defcitealias{eu.dir2015/413}{%
  {Directive ({EU}) 2015/413}}
\citealias{eu.dir2015/413}
```

(Directive (EU) 2015/413)

EU publication

- 📖 European Commission, 2015. *General report on the activities of the European Union 2014*. Luxembourg: Publications Office of the European Union.
- ⚙️ European Commission, 2015. *General report on the activities of the European Union 2014*. Luxembourg: Publications Office of the European Union.

```
@techreport{ec2015gra,
  author = {{European Commission}},
  year = {2015},
  title = {General Report on the Activities of the {European} {Union} 2014},
  address = {Luxembourg},
  publisher = {Publications Office of the European Union}}
```

HBS 37

4.10 Legal references: case reports

Legal case study

- 📖 *Seldon v Clarkson Wright & Jakes*. [2012]. UKSC 16.
- ⚙️ *Seldon v Clarkson Wright & Jakes*. [2012]. UKSC 16.

```
@jurisdiction{seldon-v-c.w.j2012,
  title = {Seldon v {Clarkson} {Wright} \& {Jakes}},
  year = {2012},
  journal = {UKSC},
  pages = {16}}
```

HBS 36

- ❗ Generally speaking, the year should be in square brackets if it is essential to the citation (unless it is a Scottish case, in which case it is printed bare), and in parentheses if it is not.
- 💡 The style assumes the year is essential if and only if a volume number is *not* provided; to override this, you can use the `year-essential` option:

```
options = {year-essential=true},
```

To use Scottish style for a case, you can either use the `scottish-style` option or the keyword `sc`.

```
options = {scottish-style},
keywords = {sc},
```

- ❗ Please bear in mind that the `options` and `keywords` fields from `biblatex` have been emulated rather crudely. You can only supply one of each and the text must match exactly (so don't add spaces or anything like that).
- 🔧 This should cover most cases, but legal references tend to enforce their own conventions no matter what the rest of the reference list is doing, and it is out of scope for this style to cater for every variation. Therefore if you need a different format (e.g. for an American case), you may prefer to format the reference more-or-less by hand:

```
title = {Seldon v {Clarkson} {Wright} \& {Jakes}},
sortyear = {2012},
note = {[2012]. UKSC 16}
```

Judgment of the European Court of Justice

 *Alessandrini Srl and others v. Commission* (C-295/03 P) [2005] ECR I-5700.
 *Alessandrini Srl and others v. Commission* (C-295/03 P) [2005] ECR I-5700.

```
@jurisdiction{srl.etal-v-comm2005,
title = {Alessandrini {Srl} and others v.\@ {Commission}},
number = {C-295/03 P},
year = {2005},
journal = {ECR},
volume = {I},
pages = {5700}}
```

HBS 39

💡 Use the `number` field (or the non-standard `casenumber` field) for the case number. For Commission Decisions, use the (non-standard) `casenumber` or (bibtex-oscola) `userb` field for the Commission case number, `number` for the formal decision number, and give ‘Commission’ as the `institution`.

References

- Adams, G., 2009. *Test citation one*. London: Imperial College Bookstall.
- Adams, G., 2014. *Test citation two*. Oxford: Oxford University Press.
- Adams, G., 2017. *Test citation three*. Cambridge: Cambridge University Press.
- Alessandrini Srl and others v. Commission* (C-295/03 P) [2005] ECR I-5700.
- Alston, S., 19 July 2004. *Society of Architectural Historians of GB*. Email to K.M. Jordan.
- Andrews, J. and Dury, A., 1773. *Map of Wiltshire*. 1 inch to 2 miles. Devizes: Wiltshire Record Society.
- Anon., 1946. *Prototype electric bicycle displayed at the ‘Britain Can Make It’ exhibition organised by the Council of Industrial Design and held at the Victoria and Albert Museum, London, 1946. Designed by B. G. Bowden* [photograph]. At: London. Design Council Slide Collection.
- The Archers*, 2006. Radio. BBC Radio 4, 23 August. 19.02 hrs.
- Aristotle, 2007. *Nicomachean ethics* (W.D. Ross. Trans.). South Dakota: NuVisions.
- Beethoven, L. van, 1950. *Symphony no.1 in C, Op.21*. Harmondsworth: Penguin.
- BS 5605:1990. *Recommendations for citing and referencing published material*. BSI.
- Burchard, J.E., 1965. How humanists use a library. In: C.F.J. Overhage and J.R. Harman, eds. *Intrex: report on a planning conference and information transfer experiments*. Cambridge, Mass.: MIT Press, pp.41–87.
- Bureau van Dijk, 2008. *BT Group plc company report. FAME* [Online]. London: Bureau van Dijk. Available from: <http://www.portal.euromonitor.com> [Accessed 6 November 2014].
- Burrell, J.G., 1973. *The importance of school tours in education*. Thesis (M.A.). Queen’s University, Belfast.
- Clark, T., 5 July 2004. A European UK Libraries Plus? *Lis-link* [Online]. Available from: lis-link@jiscmail.ac.uk [Accessed 30 July 2004].

- Crawford, G.I., 1965. Oxygen in metals. In: J.M.A. Lenihan and S.J. Thompson, eds. *Activation analysis: Proceedings of a NATO Advanced Study Institute, 2–4 August 1964 Glasgow*. London: Academic Press, pp.113–118.
- Directive (EU) 2015/413 of the European Parliament and of the Council of 11th March 2015 facilitating cross-border exchange of information on road-safety-related traffic offences [2015] *OJ* L68/9.
- Esquivel, L., 2003. *Como agua para chocolate* [Like water for chocolate]. Barcelona: Debolsillo.
- European Commission, 2015. *General report on the activities of the European Union 2014*. Luxembourg: Publications Office of the European Union.
- Great Britain. Ministry of Defence, 2004. *Delivering security in a changing world: defence white paper*. London: TSO (Cm. 6041).
- Great Britain. Parliament. House of Commons, 1988. *Local government finance bill*. London: HMSO (Bills | 1987/88, 66).
- Great Britain. Parliament. House of Commons, 2004. *National Savings investment deposits: account 2002–2003*. London: National Audit Office (HC 2003/04, 30).
- Great Britain. Parliament. House of Lords, 1987. *Social fund (Maternity and Funeral Expenses) Bill*. London: HMSO (HL 1986/87, (66)).
- Haurant, S., 2004. Britain’s borrowing hits £1 trillion. *The Guardian*, 29 July, p.16c.
- Haynes, W.M., ed., 2014. *CRC handbook of chemistry and physics* [Online]. 94th ed. Boca Raton, Fla.: CRC Press/Taylor and Francis. Available from: <http://www.hbcnetbase.com> [Accessed 16 June 2016].
- Hodds, J., 2016. *Referencing ebooks* [Kindle version 4.18]. Bath: University of Bath.
- Holland, M., 2002. *Guide to citing internet sources* [Online]. Poole: Bournemouth University. Available from: http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html [Accessed 4 November 2002].
- Hua, L., 1999. Qingdai yilai Sanxia diqu shuihan zaihai de chubu yanjiu [A preliminary study of floods and droughts in the Three Gorges region since the Qing dynasty]. *Zhongguo shehui kexue*, 1, pp.168–79.
- Hua, L. 華林甫, 1999. Qingdai yilai Sanxia diqu shuihan zaihai de chubu yanjiu 清代以來三峽地區水旱災害的初步研 [A preliminary study of floods and droughts in the Three Gorges region since the Qing dynasty]. *Zhongguo shehui kexue* 中國社會科學, 1, pp.168–79.
- The Human Medicines Regulations 2012* [Online], No.1916, United Kingdom: HMSO. Available from: http://www.legislation.gov.uk/ukxi/2012/1916/pdfs/ukxi_20121916_en.pdf [Accessed 17 April 2016].
- Iliff, D., 2006. *Royal Crescent in Bath, England - July 2006* [Online]. San Francisco: Wikimedia Foundation. Available from: https://commons.wikimedia.org/wiki/File:Royal_Crescent_in_Bath,_England_-_July_2006.jpg [Accessed 7 January 2016].
- Macbeth*, 1948. Film. Directed by Orson Welles. USA: Republic Pictures.
- The Merck index online*, n.d. [Online]. London: RSC Publishing. Available from: <http://www.rsc.org/Merck-Index> [Accessed 16 June 2016].
- Moran, C., 2016. *Save Our Libraries* [Online]. Available from: <https://www.youtube.com/watch?v=gKTfCz4JtVE&feature=youtu.be> [Accessed 29 April 2016].
- NASA, 2015. *NASA astronaut Tim Kopra on Dec. 21 spacewalk* [Online]. Washington: NASA. Available from: <http://www.nasa.gov/image-feature/nasa-astronaut-tim-kopra-on-dec-21-spacewalk> [Accessed 7 January 2015].

- Newman, R., 2010. Malaria control beyond 2010. *Brit. Med. J.*, 341(7765), pp.157–208.
- Open University, 1972. *Electricity and magnetism*. Bletchley: Open University Press.
- Oxford English dictionary*, 1989. 2nd ed. Oxford: Clarendon Press.
- Pensions Act 2014*, c.19. London: TSO.
- Phillipp Morris Inc., 1981. *Optical perforating apparatus and system*. European patent application 0021165A1. 1981-01-07.
- Rang, H.P., Dale, M.M., Ritter, J.M., Flower, R.J. and Henderson, G., 2012. *Rang and Dale's pharmacology*. 7th ed. Edinburgh: Elsevier Churchill Livingstone.
- Reid, D.R., 1967. Physical testing of polymer films. In: S.H. Pinner, ed. *Modern packaging films*. London: Butterworths, pp.143–183.
- Rick Stein's French Odyssey: Episode 5*, 2006. TV. BBC2, 23 August. 20.30 hrs.
- Rothman, K.J., Greenland, S. and Lash, T.L., eds., 2008. *Modern epidemiology*. 3rd ed. Philadelphia, Pa.: Lippincott Williams & Wilkins.
- @screencasto, n.d. *Screencast-O-Matic* (v.2) [computer program]. Available from: <https://screencast-o-matic.com/> [Accessed 16 May 2016].
- Seldon v Clarkson Wright & Jakes*. [2012]. UKSC 16.
- Shah, I. and Corrick, I., 2016. *How should central banks respond to non-neutral inflation expectations?* Bath: University of Bath. *OPUS* [Online]. Available from: <http://opus.bath.ac.uk> [Accessed 4 May 2016].
- Soper, D., 1972. Review of bracken control experiments with asulam. *Proceedings of the 11th British Weed Control Conference*, 15–17 November 1972 Brighton. Brighton: University of Sussex, pp.24–31.
- Stieg, M.F., 1981a. Continuing education and the reference librarian in the academic and research library. *Library Journal*, 105(22), pp.2547–2551.
- Stieg, M.F., 1981b. The information needs of historians. *College and Research Libraries*, 42(6), pp.549–560.
- The Independent, 1992. Picking up the bills. *The Independent*, 4 June, p.28a.
- Thurfjell, W., 1975. Vart har våran doktor tagit vägen? [Where has our doctor gone?]. *Läkartidningen*, 72, p.789.
- UNESCO, 1993. *General information programme and UNISIST*. Paris: UNESCO, (PGL-93/WS/22).
- University of Bath, 2015. *What is engineering?* [Online]. Available from: <https://www.youtube.com/watch?v=NoyZarq-Zbo> [Accessed 12 January 2016].
- Williams, F., 1997. Electronic document delivery: a trial in an academic library. *Ariadne* [Online], 10. Available from: <http://www.ariadne.ac.uk/issue10/edd/> [Accessed 5 December 1997].
- Wilson, D., 2013. *Real geometry and connectedness via triangular description: CAD example bank*. Bath: University of Bath. Available from: <https://doi.org/10.15125/BATH-00069> [Accessed 20 April 2016].
- Wiltshire Council, 2015. *Get Wiltshire walking* [Online]. Trowbridge: Wiltshire Council. Available from: <http://www.wiltshire.gov.uk/leisureandrecreation/sportphysicalactivity/getwiltshirewalking.html> [Accessed 19 August 2015].
- Witchcraft Act 1735* (9 Geo.2, c.5).

Licence

This work consists of the documented LaTeX file `bath-bst.dtx` and a `Makefile`.

The text files contained in this work may be distributed and/or modified under the conditions of the [LaTeX Project Public License \(LPPL\)](#), either version 1.3c of this license or (at your option) any later version.

This work is 'maintained' (as per LPPL maintenance status) by [Alex Ball](#).

5 Implementation

The line numbers for the listings below match `bathx.bst`.

5.1 Data model

We begin by listing the entry fields that we will recognize. These are the standard Bib_TE_X ones plus `archive`, `doi`, `eid`, `eprint`, `entrysubtype`, `eventyear`, `keywords`, `language`, `library`, `options`, `sortyear`, `titleaddon`, `translator`, `url`, `urldate`, `urlyear`, `userb`, `venue`, and `version`, all of which have been backported from `natbib` and `biblatex`. `casenumber` is specific to this style.

```
16  ENTRY
17  { address
18  archive
19  author
20  booktitle
21  casenumber
22  chapter
23  doi
24  edition
25  editor
26  eid
27  entrysubtype
28  eprint
29  eventyear
30  howpublished
31  institution
32  journal
33  key
34  keywords
35  language
36  library
37  month
38  note
39  number
40  options
41  organization
42  pages
43  publisher
44  school
45  series
46  sortyear
47  title
48  titleaddon
49  translator
50  type
51  url
52  urldate
53  urlyear
54  userb
55  venue
56  version
57  volume
58  year
59  }
60  {}
61  { label extra.label sort.label short.list }
```

5.2 Writing references out to a file

We define a set of integers for saving an output state. We will use them to to implement the mechanism for stringing the elements of the reference together. Whenever an element is printed, it is separated from any previous previous text by a delimiter, and the delimiter chosen depends on the state. We define five states which trigger increasingly strong delimiters.

```
62  INTEGERS { output.state before.all mid.clause mid.sentence after.sentence after.block }
63  FUNCTION {init.state.consts}
64  { #0 'before.all :=
65  #1 'mid.clause :=
```

```

66 #2 'mid.sentence :=
67 #3 'after.sentence :=
68 #4 'after.block :=
69 }

```

We also define two variables to store strings in.

```

70 STRINGS { s t }

```

The `output.nonnull` function removes an element from the stack, writes the appropriate delimiter to the `.tbl` file depending on the state, then restores the original element. Afterwards, the output state is set to `mid.sentence`.

- `before.all` = nothing is written.
- `mid.clause` = a space is written.
- `mid.sentence` = a comma is written.
- `after.sentence` = a period is written.
- `after.block` = a period, line break and `\newblock` are written.

```

71 FUNCTION {output.nonnull}
72 { 's :=
73   output.state mid.sentence =
74   { ", " * write$ }
75   { output.state after.sentence =
76     { add.period$ " " * write$ }
77     { output.state before.all =
78       'write$
79       { output.state mid.clause =
80         { " " * write$ }
81         { add.period$ write$
82           newline$
83           "\newblock " write$
84         }
85       }
86     }
87     if$
88   }
89   if$
90   mid.sentence 'output.state :=
91 }
92 if$
93 s
94 }

```

The ‘after’ output states persist if nothing is printed. The ‘mid.clause’ state shouldn’t, so we provide a macro for reverting to ‘mid.sentence’.

```

95 FUNCTION {end.clause}
96 { output.state mid.clause =
97   { mid.sentence 'output.state := }
98   'skip$
99   if$
100 }

```

Before writing anything out, we need to make sure there is something to be printed. We provide two functions for this: `output` is silent, while `output.check` will write a warning to the log that the field is empty. The latter consumes an additional string (at the top of the stack) for printing in the warning message.

```

101 FUNCTION {output}
102 { duplicate$ empty$
103   { pop$ end.clause }
104   'output.nonnull
105   if$
106 }
107 FUNCTION {output.check}
108 { 't :=

```

```

109 duplicate$ empty$
110 { pop$ "empty " t * " in " * cite$ * warning$
111   end.clause
112 }
113 'output.nonnull
114 if$
115 }

```

We provide some convenience functions for changing the state from `mid.sentence`. The order of precedence is `before.all`, `after.block`, `after.sentence`, `mid.clause/mid.sentence`.

```

116 FUNCTION {new.block}
117 { output.state before.all =
118   'skip$
119   { after.block 'output.state := }
120   if$
121 }
122 FUNCTION {new.sentence}
123 { output.state after.block =
124   'skip$
125   { output.state before.all =
126     'skip$
127     { after.sentence 'output.state := }
128     if$
129   }
130   if$
131 }
132 FUNCTION {continue.clause}
133 { output.state after.sentence =
134   'skip$
135   { output.state after.block =
136     'skip$
137     { output.state before.all =
138       'skip$
139       { mid.clause 'output.state := }
140       if$
141     }
142     if$
143   }
144   if$
145 }

```

The `date.block` function is a handy alias.

```

146 FUNCTION {date.block}
147 {
148   new.block
149 }

```

The `bibinfo` functions consume two tokens: the value and name of a field respectively. If the value is non-null, it is written out to the `.bbl` file in the form `\bibinfo{field name}{field value}`. If the value is missing, the tokens are replaced with an empty string; if the value is empty, it is put back on the stack. The `check` version does this silently, while the `warn` version issues warnings in the log file if the value is missing or empty.

```

150 FUNCTION {bibinfo.check}
151 { swap$
152   duplicate$ missing$
153   {
154     pop$ pop$
155     ""
156   }
157   { duplicate$ empty$
158     {
159       swap$ pop$
160     }
161     { swap$
162       "\bibinfo{" swap$ * "}-{ " * swap$ * "}" *
163     }
164     if$
165   }
166   if$

```

```

167 }
168 FUNCTION {bibinfo.warn}
169 { swap$
170   duplicate$ missing$
171   {
172     swap$ "missing " swap$ * " in " * cite$ * warning$ pop$
173     ""
174   }
175   { duplicate$ empty$
176     {
177       swap$ "empty " swap$ * " in " * cite$ * warning$
178     }
179     { swap$
180       "\bibinfo{" swap$ * "}" * swap$ * "}" *
181     }
182     if$
183   }
184   if$
185 }

```

At the end of the reference, `fin.entry` writes out a final period and line break. We will handle the start of the reference a bit later.

```

186 FUNCTION {fin.entry}
187 { add.period$
188   write$
189   newline$
190 }

```

5.3 Convenience functions

We define some useful logical operations for use with `if$`.

```

192 FUNCTION {not}
193 { { #0 }
194   { #1 }
195   if$
196 }
197 FUNCTION {and}
198 { 'skip$
199   { pop$ #0 }
200   if$
201 }
202 FUNCTION {or}
203 { { pop$ #1 }
204   'skip$
205   if$
206 }

```

`new.block.checkb` consumes two tokens and starts a new block if at least one is non-empty.

```

207 FUNCTION {new.block.checkb}
208 { empty$
209   swap$ empty$
210   and
211   'skip$
212   'new.block
213   if$
214 }

```

`field.or.null` replaces an empty (null) field value with an empty string.

```

215 FUNCTION {field.or.null}
216 { duplicate$ empty$
217   { pop$ "" }
218   'skip$
219   if$
220 }

```

The `emphasize` function applies emphasis.

```

221 FUNCTION {emphasize}
222 { duplicate$ empty$
223   { pop$ "" }
224   { "\emph{" swap$ * "}" * }
225   if$
226 }

```

The `tie.or.space.prefix` function prefixes a string with a non-breaking space if it is up to three characters long, or a normal space otherwise.

```

227 FUNCTION {tie.or.space.prefix}
228 { duplicate$ text.length$ #3 <
229   { "~" }
230   { " " }
231   if$
232   swap$
233 }

```

The `capitalize` function makes putting strings into sentence case a little more robust.

```

234 FUNCTION {capitalize}
235 { "u" change.case$ "t" change.case$ }

```

The `space.word` function adds a space to either side of a string.

```

236 FUNCTION {space.word}
237 { " " swap$ * " " * }

```

`either.or.check` takes a string and a field, and if the field is not empty, writes a warning to the log file.

```

238 FUNCTION {either.or.check}
239 { empty$
240   'pop$
241   { "can't use both " swap$ * " fields in " * cite$ * warning$ }
242   if$
243 }

```

`is.online` checks whether the entry should have ‘[Online]’ in it somewhere.

```

244 FUNCTION {is.online}
245 { type$ "software" =
246   { #0 }
247   { url empty$
248     { type$ "online" =
249       { #1 }
250       { #0 }
251       if$
252     }
253     { #1 }
254     if$
255   }
256   if$
257 }

```

5.4 Bibliography strings

Instead of dropping literal English words into the text of references, we will use these functions instead. This makes it easier to translate the style into other languages.

```

258 FUNCTION {bbl.and}
259 { "and" }
260
261 FUNCTION {bbl.etal}
262 { "et~al." }
263
264 FUNCTION {bbl.editors}
265 { "eds." }
266
267 FUNCTION {bbl.editor}
268 { "ed." }

```

```

269
270 FUNCTION {bbl.edby}
271 { "edited by" }
272
273 FUNCTION {bbl.translator}
274 { "Trans." }
275
276 FUNCTION {bbl.edition}
277 { "ed." }
278
279 FUNCTION {bbl.volume}
280 { "vol." }
281
282 FUNCTION {bbl.of}
283 { "of" }
284
285 FUNCTION {bbl.number}
286 { "no." }
287
288 FUNCTION {bbl.nr}
289 { "no." }
290
291 FUNCTION {bbl.in}
292 { "in" }
293
294 FUNCTION {bbl.pages}
295 { "pp." }
296
297 FUNCTION {bbl.page}
298 { "p." }
299
300 FUNCTION {bbl.chapter}
301 { "chap." }
302
303 FUNCTION {bbl.techrep}
304 { "Tech. Rep." }
305
306 FUNCTION {bbl.mthesis}
307 { "Master's thesis" }
308
309 FUNCTION {bbl.phdthesis}
310 { "Ph.D. thesis" }
311
312 FUNCTION {bbl.first}
313 { "1st" }
314
315 FUNCTION {bbl.second}
316 { "2nd" }
317
318 FUNCTION {bbl.third}
319 { "3rd" }
320
321 FUNCTION {bbl.fourth}
322 { "4th" }
323
324 FUNCTION {bbl.fifth}
325 { "5th" }
326
327 FUNCTION {bbl.st}
328 { "st" }
329
330 FUNCTION {bbl.nd}
331 { "nd" }
332
333 FUNCTION {bbl.rd}
334 { "rd" }
335
336 FUNCTION {bbl.th}
337 { "th" }
338
339 FUNCTION {bbl.commission}
340 { "Commission Decision" }
341

```

```

342 FUNCTION {bbl.online}
343 { "[Online]" }
344
345 FUNCTION {bbl.nodate}
346 { "n.d." }

```

These macros are codes that users can put in their .bib file instead of literal strings.

```

348 MACRO {jan} {"January"}
349
350 MACRO {feb} {"February"}
351
352 MACRO {mar} {"March"}
353
354 MACRO {apr} {"April"}
355
356 MACRO {may} {"May"}
357
358 MACRO {jun} {"June"}
359
360 MACRO {jul} {"July"}
361
362 MACRO {aug} {"August"}
363
364 MACRO {sep} {"September"}
365
366 MACRO {oct} {"October"}
367
368 MACRO {nov} {"November"}
369
370 MACRO {dec} {"December"}

```

This function allows us to convert cardinal numbers to English ordinals.

```

372 FUNCTION {eng.ord}
373 { duplicate$ "1" swap$ *
374   #-2 #1 substring$ "1" =
375   { bbl.th * }
376   { duplicate$ #-1 #1 substring$
377     duplicate$ "1" =
378     { pop$ bbl.st * }
379     { duplicate$ "2" =
380       { pop$ bbl.nd * }
381       { "3" =
382         { bbl.rd * }
383         { bbl.th * }
384         if$
385       }
386     }
387   }
388   if$
389 }
390 if$
391 }

```

5.5 Name handling

format.names consumes two tokens (field, string field name) and returns a list of names in the form 'Surname, Suffix, Initials Prefix'. To help with this, we define some additional integers and strings.

```

393 INTEGERS { nameptr namesleft numnames }
394 STRINGS { bibinfo }
395 FUNCTION {format.names}
396 { 'bibinfo :=
397   duplicate$ empty$ 'skip$ {
398     's :=
399     "" 't :=
400     #1 'nameptr :=
401     s num.names$ 'numnames :=
402     numnames 'namesleft :=

```

```

403 { namesleft #0 > }
404 { s nameptr
405   "{ll}{, jj}{, f{.}.}{-vv}"
406   format.name$
407   bibinfo bibinfo.check
408   't :=
409   nameptr #1 >
410   {
411     namesleft #1 >
412     { ", " * t * }
413     {
414       s nameptr "{ll}" format.name$ duplicate$ "others" =
415       { 't := }
416       { pop$ }
417       if$
418       t "others" =
419       {
420         " " * bbl.etal *
421       }
422       {
423         bbl.and
424         space.word * t *
425       }
426       if$
427     }
428     if$
429   }
430   't
431   if$
432   nameptr #1 + 'nameptr :=
433   namesleft #1 - 'namesleft :=
434 }
435 while$
436 } if$
437 }

```

format.names.ed consumes two tokens (field, string field name) and writes out a list of names in the form 'Initials Prefix Surname Suffix'.

```

438 FUNCTION {format.names.ed}
439 { 'bibinfo :=
440   duplicate$ empty$ 'skip$ {
441     's :=
442     "" 't :=
443     #1 'nameptr :=
444     s num.names$ 'numnames :=
445     numnames 'namesleft :=
446     { namesleft #0 > }
447     { s nameptr
448       "{f{.}.-}{vv~}{ll}{ jj}"
449       format.name$
450       bibinfo bibinfo.check
451       't :=
452       nameptr #1 >
453       {
454         namesleft #1 >
455         { ", " * t * }
456         {
457           s nameptr "{ll}" format.name$ duplicate$ "others" =
458           { 't := }
459           { pop$ }
460           if$
461           t "others" =
462           {
463             " " * bbl.etal *
464           }
465           {
466             bbl.and
467             space.word * t *
468           }
469           if$
470         }
471       }

```



```

472     if$
473     }
474     't
475     if$
476     nameptr #1 + 'nameptr :=
477     namesleft #1 - 'namesleft :=
478     }
479     while$
480 } if$
481 }

```

format.authors returns author names.

```

482 FUNCTION {format.authors}
483 { author "author" format.names
484 }

```

format.editors returns editor names, and appends 'ed.' or 'eds.' as appropriate.

```

485 FUNCTION {get.bbl.editor}
486 { editor num.names$ #1 > 'bbl.editors 'bbl.editor if$ }
487 FUNCTION {format.editors}
488 { editor "editor" format.names duplicate$ empty$ 'skip$
489   {
490     ", " *
491     " " *
492     get.bbl.editor
493     *
494   }
495   if$
496 }

```

format.translators does likewise for translators, in the form '(name. Trans.)'.

```

497 FUNCTION {format.translators}
498 { translator "translator" format.names.ed duplicate$ empty$ 'skip$
499   {
500     ". " *
501     " " *
502     bbl.translator *
503     "(" swap$ *
504     ")" *
505   }
506   if$
507 }

```

format.full.names consumes one token (a name field) and, contrary to what the name might suggest, returns a list of names in the form 'Prefix Surname'. It is called 'full' because the list is not truncated.

```

508 FUNCTION {format.full.names}
509 {'s :=
510 "" 't :=
511 #1 'nameptr :=
512 s num.names$ 'numnames :=
513 numnames 'namesleft :=
514 { namesleft #0 > }
515 { s nameptr
516   "{vv-}{ll}" format.name$
517   't :=
518   nameptr #1 >
519   {
520     namesleft #1 >
521     { ", " * t * }
522     {
523       s nameptr "{ll}" format.name$ duplicate$ "others" =
524       { 't := }
525       { pop$ }
526       if$
527       t "others" =
528       {
529         " " * bbl.etal *
530       }
531     }
532   }
533 }

```

```

531         {
532             bbl.and
533             space.word * t *
534         }
535         if$
536     }
537     if$
538 }
539 't
540 if$
541 nameptr #1 + 'nameptr :=
542 namesleft #1 - 'namesleft :=
543 }
544 while$
545 }

```

author.editor.key.full returns the author part of an author–year citation. In order of preference, it outputs the author surnames, editor surnames, the key field, or the first three characters of the entry key. It is used in book and inbook entry types.

```

546 FUNCTION {author.editor.key.full}
547 { author empty$
548   { editor empty$
549     { key empty$
550       { cite$ #1 #3 substring$ }
551       'key
552       if$
553     }
554     { editor format.full.names }
555     if$
556   }
557   { author format.full.names }
558   if$
559 }

```

editor.key.full does similar but skips the author names. It is used for the proceedings entry type.

```

560 FUNCTION {editor.key.full}
561 { editor empty$
562   { key empty$
563     { cite$ #1 #3 substring$ }
564     'key
565     if$
566   }
567   { editor format.full.names }
568   if$
569 }

```

author.key.full does similar but skips the editor names. It is used for all other entry types.

```

570 FUNCTION {author.key.full}
571 { author empty$
572   { key empty$
573     { cite$ #1 #3 substring$ }
574     'key
575     if$
576   }
577   { author format.full.names }
578   if$
579 }

```

make.full.names chooses between the above.

```

580 FUNCTION {make.full.names}
581 { type$ "book" =
582   type$ "inbook" =
583   or
584   'author.editor.key.full
585   { type$ "proceedings" =
586     'editor.key.full
587     'author.key.full
588     if$

```

```

589   }
590   if$
591 }

```

`output.bibitem` writes out the very start of the entry in the bibliography, which looks like `\bibitem[$\{$ author–year elements $\}]$ $\{$ entry key $\}$. The author–year part uses special natbib syntax: either ‘Author(Year)’ or ‘Truncated author list(Year)Full author list’. The calc.label function below supplies the truncated author list, opening parenthesis and year; you see here the closing parenthesis and conditional inclusion of the full list. The function then starts a new line and puts the entry driver into the before.all output state.`

```

592 FUNCTION {output.bibitem}
593 { newline$
594   "\bibitem[{" write$
595   label write$
596   "]" make.full.names duplicate$ short.list =
597     { pop$ }
598     { * }
599     if$
600   "]}{" * write$
601   cite$ write$
602   "]" write$
603   newline$
604   ""
605   before.all 'output.state :=
606 }

```

`format.key` takes a field (author or editor). If the field is empty, it replaces it with the key field (if not non-null). Otherwise, replaces it with an empty string. It is used as a last resort to stop an entry beginning with the date.

```

607 FUNCTION {format.key}
608 { empty$
609   { key field.or.null }
610   { "" }
611   if$
612 }

```

5.6 Titles

`select.language` checks if the field before was written out, and if so, wraps `\selectlanguage{language}\dots` around it.

```

613 FUNCTION {select.language}
614 { duplicate$ empty$
615   'pop$
616   { language empty$
617     'skip$
618     { "\selectlanguage{" language * "}" * swap$ * "}" * }
619     if$
620   }
621   if$
622 }

```

`format.title` returns the title and language fields, where the title should be in the upright shape.

```

623 FUNCTION {format.title}
624 { title
625   duplicate$ empty$ 'skip$
626   { "t" change.case$ }
627   if$
628   "title" bibinfo.check
629   duplicate$ empty$
630   'skip$
631   { select.language }
632   if$
633 }

```

`format.btitle` returns the title and language fields, where the title should be in the italic shape.

```

634 FUNCTION {format.btitle}
635 { title
636   duplicate$ empty$ 'skip$
637   { "t" change.case$ }
638   if$
639   "title" bibinfo.check
640   duplicate$ empty$ 'skip$
641   {
642     emphasize
643     select.language
644   }
645   if$
646 }

```

`format.titleaddon` returns the `titleaddon` field.

```

647 FUNCTION {format.titleaddon}
648 { titleaddon
649   duplicate$ empty$ 'skip$
650   %<bst1> { "titleaddon" bibinfo.check }
651   %<bst2> { "[" swap$ "titleaddon" bibinfo.check * "]" * }
652   if$
653 }

```

`format.version` returns the version field.

```

654 FUNCTION {format.version}
655 { version
656   duplicate$ empty$ 'skip$
657   { "(v." swap$ *
658     "version" bibinfo.check
659     ")" *
660   }
661   if$
662 }

```

`get.title.addenda` is a convenience for writing out the elements that normally follow the title in `mid.clause` state.

```

663 FUNCTION {get.title.addenda}
664 { continue.clause
665   format.version output
666   continue.clause
667   format.titleaddon output
668   continue.clause
669   %<*bst2>
670   library empty$
671   type$ "image" =
672   or
673   is.online
674   and
675   journal empty$
676   and
677   { bbl.online output }
678   'skip$
679   if$
680   continue.clause
681   %</bst2>
682   format.translators output
683 }

```

`word.in` provides the 'In:' before book titles.

```

684 FUNCTION {word.in}
685 { bbl.in capitalize
686   ":" *
687   " " * }

```

`format.booktitle` returns the booktitle field in italics.

```

688 FUNCTION {format.booktitle}
689 { booktitle
690   duplicate$ empty$ 'skip$
691   { "t" change.case$ }
692   if$
693   "booktitle" bibinfo.check
694   emphasize
695 }

```

format.in.ed.booktitle returns the booktitle, prepended by ‘In: editors, eds.’

```

696 FUNCTION {format.in.ed.booktitle}
697 { format.booktitle duplicate$ empty$ 'skip$
698   {
699     editor "editor" format.names.ed duplicate$ empty$ 'pop$
700     {
701       ", " *
702       " " *
703       get.bbl.editor
704       " " *
705       * swap$
706       *
707       word.in swap$ *
708     }
709     if$
710   }
711   if$
712 }

```

legal.journal.shape emphasizes the preceding string if it matches ‘OJ’ but otherwise does nothing.

```

713 FUNCTION {legal.journal.shape}
714 { duplicate$ "OJ" =
715   'emphasize
716   'skip$
717   if$
718 }

```

5.7 Dates

nodate.check looks at the most recent token. If it is empty, replaces it with ‘n.d.’ unless either sortyear or the nonodate option is present.

```

719 FUNCTION {nodate.check}
720 {
721   %<bst1> skip$
722   %<bst2>
723   duplicate$ empty$
724   sortyear empty$
725   and
726   { options field.or.null "nonodate" =
727     'skip$
728     { pop$ bbl.nodate }
729     if$
730   }
731   'skip$
732   if$
733   %</bst2>
734 }

```

format.date returns the year and any disambiguation label.

```

735 FUNCTION {format.date}
736 { year "year" bibinfo.check nodate.check
737   extra.label *
738 }

```

format.jur.date wraps the year and any disambiguation label in parentheses if a volume number is provided or if year-essential=false has been given as the options. If there is no volume number, or if year-essential=true has been given as the options, then the year will be

wrapped in square brackets or, if `scottish-style` has been given as the `options` or `sc` as the keywords, left bare. But if the journal is 'OJ' or 'ECR', the date is wrapped in brackets regardless. Not complicated at all, really.

```

739 FUNCTION {format.date.brackets}
740 { format.date
741   "[" swap$ * "]" *
742 }
743 FUNCTION {format.date.parens}
744 { format.date
745   "(" swap$ * ")" *
746 }
747 FUNCTION {format.jur.date.essential}
748 { options field.or.null "scottish-style" =
749   keywords field.or.null "sc" =
750   or
751     'format.date
752     'format.date.brackets
753   if$
754 }
755 FUNCTION {format.jur.date}
756 { journal field.or.null 's :=
757   s "OJ" =
758   s "ECR" =
759   or
760     'format.date.brackets
761     { options field.or.null "year-essential=false" =
762       'format.date.parens
763       { volume empty$
764         'format.jur.date.essential
765         { options field.or.null "year-essential=true" =
766           'format.jur.date.essential
767           'format.date.parens
768         if$
769       }
770     if$
771   }
772 if$
773 }
774 if$
775 }

```

`format.eu.date` wraps the year and any disambiguation label in square brackets if (and only if) the journal title is 'OJ' or 'ECR'.

```

776 FUNCTION {format.eu.date}
777 { journal field.or.null 's :=
778   s "OJ" =
779   s "ECR" =
780   or
781     'format.date.brackets
782     'format.date
783   if$
784 }

```

5.8 Series, volumes, editions

`format.bvolume` returns the volume and series information (in italics), with a comma between. A warning is given if number is also given.

```

785 FUNCTION {format.bvolume}
786 { volume empty$
787   { "" }
788   { bbl.volume volume tie.or.space.prefix
789     "volume" bibinfo.check * *
790     series "series" bibinfo.check
791     duplicate$ empty$ 'pop$
792     { emphasize ", " * swap$ * }
793     if$
794     "volume and number" number either.or.check
795   }

```

```
796   if$
797 }
```

`format.number.series` returns an empty string if a volume has been specified. Otherwise, it returns the series and the number separated by a space.

```
798 FUNCTION {format.number.series}
799 { volume empty$
800   { number empty$
801     { series field.or.null }
802     { series empty$
803       { number "number" bibinfo.check }
804       { series "series" bibinfo.check
805         number tie.or.space.prefix "number" bibinfo.check * *
806     }
807     if$
808   }
809   if$
810 }
811 { "" }
812 if$
813 }
```

`is.num` converts a string to an integer and ensures it is in the range 0–9.

```
814 FUNCTION {is.num}
815 { chr.to.int$
816   duplicate$ "0" chr.to.int$ < not
817   swap$ "9" chr.to.int$ > not and
818 }
```

`extract.num` takes a string. If it begins with an integer, it returns just that integer. Otherwise it returns the whole string.

```
819 FUNCTION {extract.num}
820 { duplicate$ 't :=
821   "" 's :=
822   { t empty$ not }
823   { t #1 #1 substring$
824     t #2 global.max$ substring$ 't :=
825     duplicate$ is.num
826     { s swap$ * 's := }
827     { pop$ "" 't := }
828   if$
829 }
830 while$
831 s empty$
832 'skip$
833 { pop$ s }
834 if$
835 }
```

`convert.edition` should operate on the edition field. If the value begins with a number (or is an English ordinal word from ‘first’ to ‘fifth’), then it returns the bibliography string version of the ordinal number (1st, 2nd, 3rd...). Otherwise, returns the edition field verbatim.

```
836 FUNCTION {convert.edition}
837 { extract.num "1" change.case$ 's :=
838   s "first" = s "1" = or
839   { bbl.first 't := }
840   { s "second" = s "2" = or
841     { bbl.second 't := }
842     { s "third" = s "3" = or
843       { bbl.third 't := }
844       { s "fourth" = s "4" = or
845         { bbl.fourth 't := }
846         { s "fifth" = s "5" = or
847           { bbl.fifth 't := }
848           { s #1 #1 substring$ is.num
849             { s eng.ord 't := }
850             { edition 't := }
851           if$
```

```

852     }
853     if$
854   }
855   if$
856   }
857   if$
858   }
859   if$
860   }
861   if$
862   t
863 }

```

`format.edition` returns the converted edition field and 'ed.'

```

864 FUNCTION {format.edition}
865 { edition duplicate$ empty$ 'skip$
866   {
867     convert.edition
868     output.state mid.sentence =
869     { "1" }
870     { "t" }
871     if$ change.case$
872     "edition" bibinfo.check
873     " " * bbl.edition *
874   }
875   if$
876 }

```

If the type is provided, `format.thesis.type` deletes the preceding string (the default value) and writes the type field with an initial capital. Otherwise it does nothing.

```

877 FUNCTION {format.thesis.type}
878 { type duplicate$ empty$
879   'pop$
880   { swap$ pop$
881     "t" change.case$ "type" bibinfo.check
882   }
883   if$
884 }

```

If the type is provided, `format.booklet.type` simply writes the type field with an initial capital. Otherwise it returns an empty string.

```

885 FUNCTION {format.booklet.type}
886 { type duplicate$ empty$
887   { pop$ "" }
888   { "type" bibinfo.check}
889   if$
890 }

```

`format.tr.number` returns the series, type and number fields as one of '(number)', '(type number)', '(series, number)', or '(series, type number)'. If a series is given, a `continue.clause` is inserted to remove the preceeding comma.

If the series begins 'HL□', it is assumed to be a House of Lords paper, thus the number acquires additional parentheses. If it equals 'C', 'Cd', 'Cmd', 'Cmnd' or 'Cm', it is followed by a dot rather than a comma (implemented as `add.cmd.number`). To help with this, `format.legr.number` consumes one token (the number) and returns the series and number as one token.

```

891 FUNCTION {add.cmd.number}
892 {
893   ". " swap$ "number" bibinfo.check *
894 }
895 FUNCTION {format.legr.number}
896 { series field.or.null 's :=
897   s "C" =
898     'add.cmd.number
899     { s "Cd" =
900       'add.cmd.number
901       { s "Cmd" =
902         'add.cmd.number
903         { s "Cmnd" =

```



```

904         'add.cmd.number
905         { s "Cm" =
906             'add.cmd.number
907             { s #1 #3 substring$ 't :=
908                 t "HL " =
909                     { ", (" swap$
910                         "number" bibinfo.check *
911                         ")" *
912                     }
913                     { ", " swap$
914                         "number" bibinfo.check *
915                     }
916                 if$
917             }
918             if$
919         }
920         if$
921     }
922     if$
923 }
924 if$
925 }
926 if$
927 }
928 FUNCTION {format.tr.number}
929 { series duplicate$ empty$
930     { pop$
931         type duplicate$ empty$
932         { pop$
933             number duplicate$ empty$
934             { pop$ ""
935             }
936             { "(" swap$
937                 "number" bibinfo.check
938                 * ")" *
939             }
940             if$
941         }
942         { "(" swap$
943             "t" change.case$ "type" bibinfo.check
944             *
945             number duplicate$ empty$
946             'pop$
947             { tie.or.space.prefix
948                 "number" bibinfo.check
949                 * *
950             }
951             if$
952             ")" *
953         }
954         if$
955     }
956     { "(" swap$ *
957         type duplicate$ empty$
958         { pop$
959             number duplicate$ empty$
960             'pop$
961             { format.legr.number * }
962             if$
963         }
964         { "t" change.case$ "type" bibinfo.check
965             *
966             number duplicate$ empty$
967             'pop$
968             { tie.or.space.prefix
969                 "number" bibinfo.check
970                 * *
971             }
972             if$
973         }
974         if$
975         ")" *
976         continue.clause

```

```

977     }
978     if$
979 }

```

`format.manual.number` returns the series, type and number fields as one of '(number)', '(type number)', '(series number)', or '(series number, type)'.

```

980 FUNCTION {format.manual.number}
981 { series duplicate$ empty$
982   { pop$
983     type duplicate$ empty$
984     { pop$
985       number duplicate$ empty$
986       { pop$ ""
987       }
988       { "number" bibinfo.check
989       }
990     }
991   }
992   { "t" change.case$ "type" bibinfo.check
993     number duplicate$ empty$
994     'pop$
995     { tie.or.space.prefix
996       "number" bibinfo.check
997       * *
998     }
999   }
1000 }
1001 if$
1002 }
1003 { number duplicate$ empty$
1004   'pop$
1005   { tie.or.space.prefix
1006     "number" bibinfo.check
1007     * *
1008   }
1009   if$
1010   type duplicate$ empty$
1011   'pop$
1012   { ", " swap$ *
1013     "t" change.case$ "type" bibinfo.check
1014     *
1015   }
1016   if$
1017 }
1018 if$
1019 }

```

`format.chapter` prints out the chapter field in the form 'c.1', for use with UK primary legislation.

```

1020 FUNCTION {format.chapter}
1021 { chapter duplicate$ empty$
1022   { pop$ "" }
1023   { "chapter" bibinfo.check
1024     "c." swap$ *
1025   }
1026   if$
1027 }

```

`format.series.chapter.number` prints out the series and chapter/number fields in the form '(series, c.1)' or '(series, number)', for use with UK legislation.

```

1028 FUNCTION {format.series.chapter.number}
1029 { series duplicate$ empty$
1030   { pop$ "" }
1031   { "series" bibinfo.check
1032     "(" swap$ *
1033     chapter duplicate$ empty$
1034     'pop$
1035     { "chapter" bibinfo.check
1036       ", c." swap$ * *
1037     }

```

```

1038     if$
1039     number duplicate$ empty$
1040     'pop$
1041     { "number" bibinfo.check
1042       ", " swap$ * *
1043     }
1044     if$
1045     ")" *
1046   }
1047 if$
1048 }

```

format.case.number returns the number field, suppressing punctuation, or inserts a period. In the case of EU Commission Decisions, the format is '(userb) Commission Decision number'

```

1049 FUNCTION {format.case.number}
1050 { casenumber duplicate$ empty$
1051   { pop$
1052     number duplicate$ empty$
1053     { pop$ "" }
1054     { institution field.or.null "Commission" =
1055       userb empty$ not
1056       and
1057         { userb "userb" bibinfo.check
1058           "(" swap$ * ")" * bbl.commission * " " *
1059           swap$ "number" bibinfo.check *
1060         }
1061         { "number" bibinfo.check
1062           "(" swap$ * ")" *
1063         }
1064       if$
1065     }
1066   if$
1067 }
1068 { "casenumber" bibinfo.check
1069   "(" swap$ * ")" *
1070   institution field.or.null "Commission" =
1071   number empty$ not
1072   and
1073   { " " * bbl.commission * " " *
1074     number "number" bibinfo.check *
1075   }
1076   'skip$
1077 if$
1078 }
1079 if$
1080 }

```

eu.case.check checks if the case is in OJ or has a number, in which case punctuation is suppressed.

```

1081 FUNCTION {eu.case.check}
1082 { journal field.or.null "OJ" =
1083   number empty$ not
1084   or
1085   casenumber empty$ not
1086   or
1087   'continue.clause
1088   'new.block
1089 if$
1090 }

```

5.9 Notes

format.note returns the note field, but changes the case of the first letter according to the output state (unless protected by braces).

```

1091 FUNCTION {format.note}
1092 { note empty$
1093   { "" }
1094   { note #1 #1 substring$

```

```

1095     duplicate$ "{" =
1096     'skip$
1097     { output.state mid.sentence =
1098     output.state mid.clause =
1099     or
1100     { "l" }
1101     { "u" }
1102     if$
1103     change.case$
1104     }
1105     if$
1106     note #2 global.max$ substring$ * "note" bibinfo.check
1107     }
1108     if$
1109     }

```

5.10 Addresses and publishers

`format.org.or.pub` intercepts a publisher (or whatever) being written out as `bibinfo`. If the write will succeed and an address has been specified, the address is written out first, then a colon and space; then the original write is allowed to proceed.

```

1110 FUNCTION {format.org.or.pub}
1111 { 't :=
1112   ""
1113   address empty$ t empty$ and
1114   'skip$
1115   {
1116     address "address" bibinfo.check *
1117     t empty$
1118     'skip$
1119     { address empty$
1120       'skip$
1121       { ": " * }
1122       if$
1123       t *
1124     }
1125     if$
1126   }
1127   if$
1128 }

```

The next three functions write out the address plus the publisher, organization and institution respectively.

```

1129 FUNCTION {format.publisher.address}
1130 { publisher "publisher" bibinfo.warn format.org.or.pub
1131 }
1132 FUNCTION {format.organization.address}
1133 { publisher empty$
1134   { organization "organization" bibinfo.check format.org.or.pub }
1135   { publisher "publisher" bibinfo.warn format.org.or.pub }
1136   if$
1137 }
1138 FUNCTION {format.institution.address}
1139 { publisher empty$
1140   { institution "institution" bibinfo.check format.org.or.pub }
1141   { publisher "publisher" bibinfo.warn format.org.or.pub }
1142   if$
1143 }

```

We add a macro for printing the library (repository) holding an eprint.

```

1144 FUNCTION {format.library}
1145 { library "library" bibinfo.check
1146   emphasize
1147   %< *bst2 >
1148   duplicate$ empty$ not
1149   is.online
1150   and
1151   journal empty$

```

```

1152 and
1153 { " " * bbl.online * }
1154 'skip$
1155 if$
1156 %</bst2>
1157 }

```

The library field is also used for where a physical image is kept. As we have only one address field, it will attach to whichever is present out of organization, publisher and library. In case there is more than one, it will attach to publisher in preference to library and library in preference to organization.

```

1158 FUNCTION {format.img.library}
1159 { library duplicate$ empty$
1160   { pop$ "" }
1161   { "library" bibinfo.check
1162     "At: " swap$ *
1163   }
1164   if$
1165 }
1166 FUNCTION {format.img.library.address}
1167 { address empty$ library empty$ and institution empty$ and
1168   { "" }
1169   { "At: "
1170     address "address" bibinfo.check *
1171     library duplicate$ empty$
1172     { institution duplicate$ empty$
1173       { pop$ "" }
1174       { address empty$
1175         'skip$
1176         { ". " swap$ }
1177         if$
1178         "institution" bibinfo.check * *
1179       }
1180       if$
1181     }
1182     { address empty$
1183       'skip$
1184       { ". " swap$ }
1185       if$
1186       "library" bibinfo.check * *
1187     }
1188     if$
1189   }
1190   if$
1191 }
1192 FUNCTION {format.pub.org.lib.address}
1193 { publisher duplicate$ empty$
1194   { pop$
1195     organization duplicate$ empty$
1196     { pop$
1197       format.img.library.address }
1198     { "organization" bibinfo.check
1199       library empty$
1200       { format.org.or.pub }
1201       { ". " *
1202         format.img.library.address *
1203       }
1204       if$
1205     }
1206     if$
1207   }
1208   { "publisher" bibinfo.warn format.org.or.pub
1209     library empty$
1210     'skip$
1211     { ". " *
1212       format.img.library *
1213     }
1214     if$
1215   }
1216   if$
1217 }

```

5.11 Pinpointing

`multi.page.check` should operate on the `pages` field. If the value is a range, that is, contains '-', ', or '+', returns 1, otherwise returns 0.

```
1219 INTEGERS { multiresult }
1220 FUNCTION {multi.page.check}
1221 { 't :=
1222   #0 'multiresult :=
1223   { multiresult not
1224     t empty$ not
1225     and
1226   }
1227   { t #1 #1 substring$
1228     duplicate$ "-" =
1229     swap$ duplicate$ "," =
1230     swap$ "+" =
1231     or or
1232     { #1 'multiresult := }
1233     { t #2 global.max$ substring$ 't := }
1234     if$
1235   }
1236   while$
1237   multiresult
1238 }
```

`n.dashify` converts hyphens in page ranges into en dashes.

```
1239 FUNCTION {n.dashify}
1240 {
1241   't :=
1242   ""
1243   { t empty$ not }
1244   { t #1 #1 substring$ "-" =
1245     { t #1 #2 substring$ "--" = not
1246       { "--" *
1247         t #2 global.max$ substring$ 't :=
1248       }
1249       { { t #1 #1 substring$ "-" = }
1250         { "-" *
1251           t #2 global.max$ substring$ 't :=
1252         }
1253         while$
1254       }
1255     }
1256   }
1257   { t #1 #1 substring$ *
1258     t #2 global.max$ substring$ 't :=
1259   }
1260   if$
1261 }
1262 while$
1263 }
```

`format.pages` returns the `pages` fields, prefixed by 'p.' or 'pp.' as appropriate.

```
1264 FUNCTION {format.pages}
1265 { pages duplicate$ empty$ 'skip$
1266   { duplicate$ multi.page.check
1267     {
1268       bbl.pages swap$
1269       n.dashify
1270     }
1271     {
1272       bbl.page swap$
1273     }
1274     if$
1275     "pages" bibinfo.check
1276     *
1277   }
1278   if$
1279 }
```

`format.journal.pages` does similar, but it looks at the preceding string and, if both are non-empty, inserts a comma between it and the pages string.

```

1280 FUNCTION {format.journal.pages}
1281 { pages duplicate$ empty$ 'pop$
1282   { swap$ duplicate$ empty$
1283     { pop$ pop$ format.pages }
1284     {
1285       ", " *
1286       swap$
1287       n.dashify
1288       pages multi.page.check
1289       'bbl.pages
1290       'bbl.page
1291       if$
1292       swap$
1293       "pages" bibinfo.check
1294       * *
1295     }
1296   }
1297 }
1298 if$
1299 }

```

`format.journal.eid` returns the EID, but it looks at the preceding string and, if both are non-empty, inserts a comma between it and the EID string. The EID is a code used by some online-only journals to identify an article within a volume or issue instead of a page range (since the page numbering resets for each article, not each volume/issue).

```

1300 FUNCTION {format.journal.eid}
1301 { eid "eid" bibinfo.check
1302   duplicate$ empty$ 'pop$
1303   { swap$ duplicate$ empty$ 'skip$
1304     {
1305       ", " *
1306     }
1307     if$
1308     swap$ *
1309   }
1310   if$
1311 }

```

`format.vol.num.pages` returns the journal pinpointing in the form 'volume(number), eid/pages'. It writes a warning to the log if `number` is given instead of `volume`: if a journal does not organize itself into volumes, treat every issue as its own volume!

```

1312 FUNCTION {format.vol.num.pages}
1313 { volume field.or.null
1314   duplicate$ empty$ 'skip$
1315   {
1316     "volume" bibinfo.check
1317   }
1318   if$
1319   number "number" bibinfo.check duplicate$ empty$ 'skip$
1320   {
1321     swap$ duplicate$ empty$
1322     { "there's a number but no volume in " cite$ * warning$ }
1323     'skip$
1324     if$
1325     swap$
1326     "(" swap$ * ")" *
1327   }
1328   if$ *
1329   eid empty$
1330   { format.journal.pages }
1331   { format.journal.eid }
1332   if$
1333 }

```

`format.chapter.pages` returns pinpointing information for books, in the form 'chap. chapter, p.pages'.

```

1334 FUNCTION {format.chapter.pages}
1335 { chapter empty$
1336   'format.pages
1337   { type empty$
1338     { bbl.chapter }
1339     { type "1" change.case$
1340       "type" bibinfo.check
1341     }
1342     if$
1343     chapter tie.or.space.prefix
1344     "chapter" bibinfo.check
1345     * *
1346     pages empty$
1347     'skip$
1348     { ", " * format.pages * }
1349     if$
1350   }
1351   if$
1352 }

```

format.journal.series.vol.pages returns the journal title and pinpointing in the form 'volume journal pages', 'journal volume-pages' or 'journal series volume/pages', depending on the journal.

```

1353 FUNCTION {format.journal.series.vol.pages}
1354 { journal field.or.null
1355   duplicate$ "QJ" =
1356   { "journal" bibinfo.check
1357     emphasize
1358     series empty$
1359     volume empty$
1360     and
1361     pages empty$
1362     and
1363     eid empty$
1364     and
1365     'skip$
1366     { " " * }
1367     if$
1368     series "series" bibinfo.check
1369     volume "volume" bibinfo.check *
1370     duplicate$ empty$ pages empty$ and
1371     'skip$
1372     { "/" * }
1373     if$ *
1374   }
1375   { duplicate$ "ECR" =
1376     { "journal" bibinfo.check
1377       volume "volume" bibinfo.check
1378       duplicate$ empty$
1379       'skip$
1380       { " " swap$ * "--" * * }
1381       if$
1382     }
1383     { "journal" bibinfo.check
1384     %< *bst2>
1385       is.online
1386       { " " * bbl.online * }
1387       'skip$
1388       if$
1389     %< /bst2>
1390     volume "volume" bibinfo.check
1391     duplicate$ empty$
1392     'pop$
1393     { " " * swap$ * }
1394     if$
1395     pages empty$
1396     eid empty$
1397     and
1398     'skip$
1399     { " " * }
1400     if$

```



```

1401     }
1402     if$
1403   }
1404   if$
1405   eid empty$
1406     { pages "pages" bibinfo.check * }
1407     { eid "eid" bibinfo.check * }
1408   if$
1409 }

```

5.12 URLs

`format.eprint` constructs a string `\eprint[archive]{eprint}`.

```

1410 FUNCTION {format.eprint}
1411 { eprint duplicate$ empty$
1412   'skip$
1413   { "\eprint"
1414     archive empty$
1415     'skip$
1416     { "[" * archive * "]" * }
1417     if$
1418     "{" * swap$ * "}" *
1419   }
1420   if$
1421 }

```

`format.url` constructs a string `Available from: \url{url} [Accessed urldate]`.

```

1422 FUNCTION {format.url}
1423 { doi empty$
1424   { url }
1425   { "https://doi.org/" doi * }
1426   if$
1427   duplicate$ empty$
1428     { pop$ "" }
1429     { "\urlprefix\url{" swap$ * "}" * }
1430   if$
1431   urlyear empty$
1432     { urldate empty$
1433       'skip$
1434       { "[\urldateprefix{}]" * urldate * "}" * }
1435     if$
1436   }
1437   { " [\urldateprefix{}]" * urlyear * "}" * }
1438   if$
1439 }

```

5.13 Cross referencing between entries

The following functions are used for entries that look like ‘Author, (Year). Title. In: Author (Year).’ In practice, they should never be used with Harvard (Bath).

```

1440 FUNCTION {format.article.crossref}
1441 {
1442   word.in
1443   " \cite{" * crossref * "}" *
1444 }
1445 FUNCTION {format.incoll.inproc.crossref}
1446 {
1447   word.in
1448   " \cite{" * crossref * "}" *
1449 }

```

`format.book.crossref` is a bit different as it is intended for cases where the given book is volume X of a larger work: ‘Author, (Year). Title. Vol. volume of Author (Year).’ In the absence of a volume, writes a warning to the log and behaves like the above.

```

1450 FUNCTION {format.book.crossref}
1451 { volume duplicate$ empty$
1452   { "empty volume in " cite$ * "'s crossref of " * crossref * warning$
1453     pop$ word.in
1454   }
1455   { bbl.volume
1456     capitalize
1457     swap$ tie.or.space.prefix "volume" bibinfo.check * * bbl.of space.word *
1458   }
1459   if$
1460   " \cite{" * crossref * "}" *
1461 }

```

5.14 Formatting drivers for specific entry types

These functions are called when an entry of the given type is being formatted.

5.14.1 Article

```

1463 FUNCTION {article}
1464 { output.bibitem
1465   format.authors "author" output.check
1466   author format.key output
1467   format.date "year" output.check
1468   date.block
1469   format.title "title" output.check
1470   get.title.addenda
1471   new.block
1472   crossref missing$
1473   {
1474     journal
1475     "journal" bibinfo.check
1476     emphasize
1477     "journal" output.check
1478     %<bst2>
1479     is.online
1480     { continue.clause
1481       bbl.online output
1482     }
1483     'skip$
1484     if$
1485     %</bst2>
1486     format.vol.num.pages output
1487   }
1488   { format.article.crossref output.nonnull
1489     format.pages output
1490   }
1491   if$
1492   new.block
1493   format.note output
1494   new.block
1495   format.eprint output
1496   format.url output
1497   fin.entry
1498 }

```

5.14.2 Book

```

1499 FUNCTION {book}
1500 { output.bibitem
1501   author empty$
1502   { editor empty$
1503     { format.btitle "title" output.check
1504       format.date "year" output.check
1505       year empty$ 'skip$ { date.block } if$
1506       get.title.addenda
1507     }
1508     { format.editors "author and editor" output.check
1509       editor format.key output

```

```

1510     format.date "year" output.check
1511     date.block
1512     format.btitle "title" output.check
1513     get.title.addenda
1514   }
1515   if$
1516 }
1517 { format.authors output.nonnull
1518   crossref missing$
1519   { "author and editor" editor either.or.check }
1520   'skip$
1521   if$
1522     format.date "year" output.check
1523     date.block
1524     format.btitle "title" output.check
1525     get.title.addenda
1526   }
1527 if$
1528 format.bvolume output
1529 format.number.series output
1530 new.block
1531 format.edition output
1532 crossref missing$
1533 { new.sentence
1534   format.publisher.address output
1535 }
1536 {
1537   new.block
1538   format.book.crossref output.nonnull
1539 }
1540 if$
1541 new.block
1542 format.note output
1543 new.block
1544 format.eprint output
1545 format.url output
1546 fin.entry
1547 }

```

5.14.3 Unpublished or self-published monograph

```

1548 FUNCTION {booklet}
1549 { output.bibitem
1550   author empty$
1551   { format.btitle "title" output.check
1552     format.date "year" output.check
1553     year empty$ 'skip$ { date.block } if$
1554     get.title.addenda
1555   }
1556   { format.authors output
1557     author format.key output
1558     format.date "year" output.check
1559     date.block
1560     format.btitle "title" output.check
1561     get.title.addenda
1562   }
1563   if$
1564   new.block
1565   format.booklet.type output
1566   new.block
1567   howpublished "howpublished" bibinfo.check output
1568   new.block
1569   format.note output
1570   publisher empty$ 'skip$
1571   { new.block}
1572   if$
1573   format.publisher.address output
1574   new.block
1575   format.eprint output
1576   format.url output
1577   fin.entry

```

```
1578 }
```

5.14.4 Work in book

```
1579 FUNCTION {inbook}
1580 { output.bibitem
1581   author empty$
1582   { format.editors "author and editor" output.check
1583     editor format.key output
1584   }
1585   { format.authors output.nonnull
1586     crossref missing$
1587     { "author and editor" editor either.or.check }
1588     'skip$
1589     if$
1590   }
1591   if$
1592   format.date "year" output.check
1593   date.block
1594   format.btitle "title" output.check
1595   get.title.addenda
1596   crossref missing$
1597   {
1598     format.publisher.address output
1599     format.bvolume output
1600     format.chapter.pages "chapter and pages" output.check
1601     new.block
1602     format.number.series output
1603     new.sentence
1604   }
1605   {
1606     format.chapter.pages "chapter and pages" output.check
1607     new.block
1608     format.book.crossref output.nonnull
1609   }
1610   if$
1611   format.edition output
1612   new.block
1613   format.note output
1614   new.block
1615   format.eprint output
1616   format.url output
1617   fin.entry
1618 }
```

5.14.5 Work in collection

```
1619 FUNCTION {incollection}
1620 { output.bibitem
1621   format.authors "author" output.check
1622   author format.key output
1623   format.date "year" output.check
1624   date.block
1625   format.title "title" output.check
1626   get.title.addenda
1627   new.block
1628   crossref missing$
1629   { format.in.ed.booktitle "booktitle" output.check
1630     new.block
1631     format.publisher.address output
1632     format.bvolume output
1633     format.number.series output
1634     format.chapter.pages output
1635     new.sentence
1636     format.edition output
1637   }
1638   { format.incoll.inproc.crossref output.nonnull
1639     format.chapter.pages output
1640   }
1641   if$
```

```

1642 new.block
1643 format.note output
1644 new.block
1645 format.eprint output
1646 format.url output
1647 fin.entry
1648 }

```

5.14.6 Conference paper

```

1649 FUNCTION {inproceedings}
1650 { output.bibitem
1651   format.authors "author" output.check
1652   author format.key output
1653   format.date "year" output.check
1654   date.block
1655   format.title "title" output.check
1656   get.title.addenda
1657   new.block
1658   crossref missing$
1659   { format.in.ed.booktitle "booktitle" output.check
1660     eventyear output
1661     continue.clause
1662     venue output
1663     new.sentence
1664     publisher empty$
1665     { format.organization.address output }
1666     { organization "organization" bibinfo.check output
1667       format.publisher.address output
1668     }
1669     if$
1670     format.bvolume output
1671     format.number.series output
1672     format.pages output
1673   }
1674   { format.incoll.inproc.crossref output.nonnull
1675     format.pages output
1676   }
1677   if$
1678   new.block
1679   format.note output
1680   new.block
1681   format.eprint output
1682   format.url output
1683   fin.entry
1684 }
1685 FUNCTION {conference} { inproceedings }

```

5.14.7 Legal case studies

```

1686 FUNCTION {jurisdiction}
1687 { output.bibitem
1688   author empty$
1689   { format.btitle "title" output.check
1690     continue.clause
1691     format.case.number output
1692     number empty$
1693     casenumber empty$
1694     and
1695     'new.sentence
1696     'continue.clause
1697     if$
1698     format.jur.date "year" output.check
1699     year empty$ 'skip$ { eu.case.check } if$
1700     get.title.addenda
1701   }
1702   { format.authors output
1703     author format.key output
1704     format.jur.date "year" output.check

```

```

1705     date.block
1706     continue.clause
1707     format.btitle "title" output.check
1708     get.title.addenda
1709   }
1710   if$
1711   eu.case.check
1712   format.note output
1713   note empty$
1714   'eu.case.check
1715   'new.block
1716   if$
1717   crossref missing$
1718     { continue.clause
1719       format.journal.series.vol.pages output
1720     }
1721     { format.article.crossref output.nonnull
1722       format.pages output
1723     }
1724   if$
1725   new.block
1726   format.eprint output
1727   format.url output
1728   fin.entry
1729 }

```

5.14.8 Legislation

```

1730 FUNCTION {uklegislation}
1731 { output.bibitem
1732   author empty$
1733   { format.btitle "title" output.check
1734     continue.clause
1735     format.date emphasize "year" output.check
1736     get.title.addenda
1737   }
1738   { format.authors output
1739     author format.key output
1740     format.date "year" output.check
1741     date.block
1742     continue.clause
1743     format.btitle "title" output.check
1744     get.title.addenda
1745   }
1746   if$
1747   entrysubtype field.or.null "secondary" =
1748     { number "number" bibinfo.check output }
1749     'skip$
1750   if$
1751   series empty$
1752     { format.chapter output }
1753     'skip$
1754   if$
1755   format.note output
1756   publisher empty$ 'skip$
1757     { entrysubtype field.or.null "secondary" =
1758       'skip$
1759       { new.block }
1760     if$
1761     format.publisher.address output
1762   }
1763   if$
1764   continue.clause
1765   format.series.chapter.number output
1766   new.block
1767   format.eprint output
1768   format.url output
1769   fin.entry
1770 }
1771 FUNCTION {eulegislation}
1772 { output.bibitem

```

```

1773 author empty$
1774 { format.title "title" output.check
1775   continue.clause
1776   format.eu.date "year" output.check
1777   get.title.addenda
1778 }
1779 { format.authors output
1780   author format.key output
1781   format.eu.date "year" output.check
1782   date.block
1783   continue.clause
1784   format.title "title" output.check
1785   get.title.addenda
1786 }
1787 if$
1788 eu.case.check
1789 format.note output
1790 note empty$
1791 'eu.case.check
1792 'new.block
1793 if$
1794 crossref missing$
1795   { continue.clause
1796     format.journal.series.vol.pages output
1797   }
1798   { format.article.crossref output.nonnull
1799     format.pages output
1800   }
1801 if$
1802 new.block
1803 format.eprint output
1804 format.url output
1805 fin.entry
1806 }
1807 FUNCTION {legislation}
1808 { journal empty$
1809   'uklegislation
1810   'eulegislation
1811 if$
1812 }

```

5.14.9 Manual

```

1813 FUNCTION {manual}
1814 { output.bibitem
1815   author empty$
1816   { format.btitle "title" output.check
1817     format.date "year" output.check
1818     year empty$ 'skip$ { date.block } if$
1819     get.title.addenda
1820   }
1821   { format.authors output
1822     author format.key output
1823     format.date "year" output.check
1824     date.block
1825     format.btitle "title" output.check
1826     get.title.addenda
1827   }
1828 if$
1829 format.edition output
1830 new.block
1831 format.manual.number output
1832 new.block
1833 format.library output
1834 organization address new.block.checkb
1835 format.organization.address output
1836 new.block
1837 format.note output
1838 new.block
1839 format.eprint output
1840 format.url output

```

```
1841   fin.entry
1842 }
```

5.14.10 Image

```
1843 FUNCTION {image}
1844 { output.bibitem
1845   author empty$
1846   { format.btitle "title" output.check
1847     format.date "year" output.check
1848     year empty$ 'skip$ { date.block } if$
1849     get.title.addenda
1850   }
1851   { format.authors output
1852     author format.key output
1853     format.date "year" output.check
1854     date.block
1855     format.btitle "title" output.check
1856     get.title.addenda
1857   }
1858   if$
1859   format.edition output
1860   new.block
1861   format.manual.number output
1862   new.block
1863   format.pub.org.lib.address output
1864   new.block
1865   format.note output
1866   new.block
1867   format.eprint output
1868   format.url output
1869   fin.entry
1870 }
```

5.14.11 Master's thesis

```
1871 FUNCTION {mastersthesis}
1872 { output.bibitem
1873   format.authors "author" output.check
1874   author format.key output
1875   format.date "year" output.check
1876   date.block
1877   format.btitle "title" output.check
1878   get.title.addenda
1879   new.block
1880   bbl.mthesis format.thesis.type output.nonnull
1881   new.block
1882   school "school" bibinfo.warn output
1883   address "address" bibinfo.check output
1884   new.block
1885   format.note output
1886   new.block
1887   format.eprint output
1888   format.url output
1889   fin.entry
1890 }
```

5.14.12 Miscellaneous

```
1891 FUNCTION {misc}
1892 { output.bibitem
1893   author empty$
1894   { format.title "title" output.check
1895     format.date "year" output.check
1896     year empty$ 'skip$ { date.block } if$
1897     get.title.addenda
1898   }
1899   { format.authors output
```



```

1900     author format.key output
1901     format.date "year" output.check
1902     date.block
1903     format.title "title" output.check
1904     get.title.addenda
1905   }
1906   if$
1907   new.block
1908   howpublished "howpublished" bibinfo.check output
1909   new.block
1910   format.note output
1911   new.block
1912   format.eprint output
1913   format.url output
1914   fin.entry
1915 }

```

5.14.13 Doctoral thesis

```

1916 FUNCTION {phdthesis}
1917 { output.bibitem
1918   format.authors "author" output.check
1919   author format.key output
1920   format.date "year" output.check
1921   date.block
1922   format.btitle "title" output.check
1923   get.title.addenda
1924   new.block
1925   bbl.phdthesis format.thesis.type output.nonnull
1926   new.block
1927   school "school" bibinfo.warn output
1928   address "address" bibinfo.check output
1929   new.block
1930   format.note output
1931   new.block
1932   format.eprint output
1933   format.url output
1934   fin.entry
1935 }

```

5.14.14 Conference proceedings

```

1936 FUNCTION {proceedings}
1937 { output.bibitem
1938   editor empty$
1939   { format.btitle "title" output.check
1940     format.date "year" output.check
1941     year empty$ 'skip$ { date.block } if$
1942     get.title.addenda
1943   }
1944   { format.editors output
1945     editor format.key output
1946     format.date "year" output.check
1947     date.block
1948     format.btitle "title" output.check
1949     get.title.addenda
1950   }
1951   if$
1952   format.bvolume output
1953   format.number.series output
1954   new.sentence
1955   publisher empty$
1956   { format.organization.address output }
1957   { organization "organization" bibinfo.check output
1958     format.publisher.address output
1959   }
1960   if$
1961   new.block
1962   format.note output
1963   new.block

```

```

1964 format.eprint output
1965 format.url output
1966 fin.entry
1967 }

```

5.14.15 Report

```

1968 FUNCTION {techreport}
1969 { output.bibitem
1970   author empty$
1971   { format.btitle "title" output.check
1972     format.date "year" output.check
1973     year empty$ 'skip$ { date.block } if$
1974     get.title.addenda
1975   }
1976   { format.authors output
1977     author format.key output
1978     format.date "year" output.check
1979     date.block
1980     format.btitle "title" output.check
1981     get.title.addenda
1982   }
1983   if$
1984   new.block
1985   publisher empty$
1986   { format.institution.address output }
1987   { institution "institution" bibinfo.check output
1988     format.publisher.address output
1989   }
1990   if$
1991   format.tr.number output
1992   new.block
1993   format.library output
1994   new.block
1995   format.note output
1996   new.block
1997   format.eprint output
1998   format.url output
1999   fin.entry
2000 }

```

5.14.16 Unpublished works

```

2001 FUNCTION {unpublished}
2002 { output.bibitem
2003   author empty$
2004   { format.btitle "title" output.check
2005     format.date "year" output.check
2006     year empty$ 'skip$ { date.block } if$
2007     get.title.addenda
2008   }
2009   { format.authors output
2010     author format.key output
2011     format.date "year" output.check
2012     date.block
2013     format.btitle "title" output.check
2014     get.title.addenda
2015   }
2016   if$
2017   new.block
2018   howpublished "howpublished" bibinfo.check output
2019   new.block
2020   format.note output
2021   new.block
2022   format.eprint output
2023   format.url output
2024   fin.entry
2025 }

```

5.14.17 Aliases and fallbacks

Here are a list of aliases supported by this style.

```
2027 FUNCTION {letter} {
2028   journal empty$
2029   'unpublished
2030   'article
2031   if$
2032 }
2033 FUNCTION {audio} { booklet }
2034 FUNCTION {movie} { booklet }
2035 FUNCTION {music} { booklet }
2036 FUNCTION {video} { booklet }
2037 FUNCTION {dataset} { manual }
2038 FUNCTION {electronic} { manual }
2039 FUNCTION {online} { manual }
2040 FUNCTION {patent} { manual }
2041 FUNCTION {software} { manual }
2042 FUNCTION {standard} { manual }
2043 FUNCTION {www} { manual }
2044 FUNCTION {thesis} { phdthesis }
2045 FUNCTION {report} { techreport }
```

The fallback driver is `misc`, as with most styles.

```
2046 FUNCTION {default.type} { misc }
```

5.15 Bibliography processing

We are now ready for Bib_T_EX to read in the `.bib` file.

```
2048 READ
```

These functions are used to generate a sort key for each entry in the `.bib` file. `sortify` strips out \LaTeX commands, spaces and ASCII symbols, and converts to lowercase.

```
2049 FUNCTION {sortify}
2050 { purify$
2051   "l" change.case$
2052 }
```

`chop.word` takes a string `s`, an integer `len`, and a prefix (stop) string. If the first `len` characters of `s` match the prefix, those characters are stripped off `s`.

```
2053 INTEGERS { len }
2054 FUNCTION {chop.word}
2055 { 's :=
2056   'len :=
2057   s #1 len substring$ =
2058     { s len #1 + global.max$ substring$ }
2059   's
2060   if$
2061 }
```

`format.lab.names` is similar to `format.full.names`, the main difference being that if there are more than 3 names in the list, the second and subsequent names are replaced with 'et al.'

```
2062 FUNCTION {format.lab.names}
2063 {'s :=
2064 "" 't :=
2065 #1 'nameptr :=
2066 s num.names$ 'numnames :=
2067 numnames 'namesleft :=
2068 { namesleft #0 > }
2069 { s nameptr
2070   "{vv-}{ll}" format.name$
2071   't :=
2072   nameptr #1 >
2073   {
2074     nameptr #2 =
2075     numnames #3 > and
```

```

2076     { "others" 't :=
2077     #1 'namesleft := }
2078     'skip$
2079     if$
2080     namesleft #1 >
2081     { ", " * t * }
2082     {
2083     s nameptr "{ll}" format.name$ duplicate$ "others" =
2084     { 't := }
2085     { pop$ }
2086     if$
2087     t "others" =
2088     {
2089     " " * bbl.etal *
2090     }
2091     {
2092     bbl.and
2093     space.word * t *
2094     }
2095     if$
2096     }
2097     if$
2098     }
2099     't
2100     if$
2101     nameptr #1 + 'nameptr :=
2102     namesleft #1 - 'namesleft :=
2103     }
2104     while$
2105     }

```

author.editor.key.label is the analogue of author.editor.key.full but with a truncated list of author/editor names.

```

2106     FUNCTION {author.editor.key.label}
2107     { author empty$
2108     { editor empty$
2109     { key empty$
2110     { title empty$
2111     { cite$ #1 #3 substring$ }
2112     { title
2113     type$ "book" =
2114     { emphasize }
2115     'skip$
2116     if$
2117     }
2118     if$
2119     }
2120     'key
2121     if$
2122     }
2123     { editor format.lab.names }
2124     if$
2125     }
2126     { author format.lab.names }
2127     if$
2128     }

```

editor.key.label is the analogue of editor.key.full but with a truncated list of editor names.

```

2129     FUNCTION {editor.key.label}
2130     { editor empty$
2131     { key empty$
2132     { title empty$
2133     { cite$ #1 #3 substring$ }
2134     { title emphasize }
2135     if$
2136     }
2137     'key
2138     if$
2139     }
2140     { editor format.lab.names }

```

```
2141 if$
2142 }
```

author.key.label is the analogue of author.key.full but with a truncated list of author names.

```
2143 FUNCTION {author.key.label}
2144 { author empty$
2145   { key empty$
2146     { title empty$
2147       { cite$ #1 #3 substring$ }
2148       { title
2149         type$ "article" =
2150         type$ "incollection" =
2151         or
2152         type$ "inproceedings" =
2153         or
2154         type$ "misc" =
2155         or
2156         'skip$
2157         { emphasize }
2158         if$
2159       }
2160     }
2161   }
2162   'key
2163   if$
2164 }
2165 { author format.lab.names }
2166 if$
2167 }
```

calc.short.authors chooses between the above.

```
2168 FUNCTION {calc.short.authors}
2169 { type$ "book" =
2170   type$ "inbook" =
2171   or
2172   'author.editor.key.label
2173   { type$ "proceedings" =
2174     'editor.key.label
2175     'author.key.label
2176     if$
2177   }
2178   if$
2179   'short.list :=
2180 }
```

calc.label creates a string in the form ‘Truncated Author List(Year’. The closing parenthesis is supplied by output.bibitem (see above), into which this string is incorporated.

```
2181 FUNCTION {calc.label}
2182 { calc.short.authors
2183   short.list
2184   "("
2185   *
2186   year nodate.check duplicate$ empty$
2187   { pop$ "" }
2188   'skip$
2189   if$
2190   *
2191   'label :=
2192 }
```

sort.format.names consumes one token (a field) and constructs a list of names in the form ‘Surname Forename Prefix Suffix’, with just spaces between adjacent names and ‘zzzzz’ used in place of ‘et al.’

```
2193 FUNCTION {sort.format.names}
2194 { 's :=
2195   #1 'nameptr :=
2196   ""
2197   s num.names$ 'numnames :=
```

```

2198 numnames 'namesleft :=
2199 { namesleft #0 > }
2200 { s nameptr
2201 "{ll{ }}{ f{ }}{ jj{ }}{ vv{ }}"
2202 format.name$ 't :=
2203 nameptr #1 >
2204 {
2205     " " *
2206     namesleft #1 =
2207     t "others" =
2208     and
2209     { "zzzzz" 't := }
2210     'skip$
2211     if$
2212     t sortify *
2213 }
2214 { t sortify * }
2215 if$
2216 nameptr #1 + 'nameptr :=
2217 namesleft #1 - 'namesleft :=
2218 }
2219 while$
2220 }

```

sort.format.title strips off initial articles from the preceding field and normalizes what remains using sortify (see above).

```

2221 FUNCTION {sort.format.title}
2222 { 't :=
2223 "A " #2
2224 "An " #3
2225 "The " #4 t chop.word
2226 chop.word
2227 chop.word
2228 sortify
2229 #1 global.max$ substring$
2230 }

```

author.editor.sort creates a string for use when sorting the entry in the bibliography. In order of preference, the string is based on author, editor, title or the key field.

```

2231 FUNCTION {author.editor.sort}
2232 { author empty$
2233   { editor empty$
2234     { title empty$
2235       { key empty$
2236         { "to sort, need author, editor, title, or key in " cite$ * warning$
2237           ""
2238         }
2239         { key sortify }
2240       if$
2241     }
2242     { title sort.format.title }
2243   if$
2244 }
2245 { editor sort.format.names }
2246 if$
2247 }
2248 { author sort.format.names }
2249 if$
2250 }

```

editor.sort does similar but skips the author names.

```

2251 FUNCTION {editor.sort}
2252 { editor empty$
2253   { title empty$
2254     { key empty$
2255       { "to sort, need editor, title, or key in " cite$ * warning$
2256         ""
2257       }
2258       { key sortify }
2259     if$

```

```

2260     }
2261     { title sort.format.title }
2262     if$
2263   }
2264   { editor sort.format.names }
2265   if$
2266 }

```

author.sort does similar but skips the editor names.

```

2267 FUNCTION {author.sort}
2268 { author empty$
2269   { title empty$
2270     { key empty$
2271       { "to sort, need author, title, or key in " cite$ * warning$
2272         ""
2273       }
2274       { key sortify }
2275     }
2276   }
2277   { title sort.format.title }
2278   if$
2279 }
2280 { author sort.format.names }
2281 if$
2282 }

```

presort constructs the sort.label from the calc.label and one of the above three functions. It also constructs the sort.key\$ (used by SORT) from the sort.label and the title.

```

2284 FUNCTION {presort}
2285 { calc.label
2286   label sortify
2287   " "
2288   *
2289   type$ "book" =
2290   type$ "inbook" =
2291   or
2292   'author.editor.sort
2293   { type$ "proceedings" =
2294     'editor.sort
2295     'author.sort
2296   }
2297   if$
2298   #1 entry.max$ substring$
2299   'sort.label :=
2300   sort.label
2301   *
2302   " "
2303   *
2304   title field.or.null
2305   sort.format.title
2306   *
2307   #1 entry.max$ substring$
2308   'sort.key$ :=
2309 }
2310 }

```

We now tell BibTEX to go through each entry in turn and apply the presort function, then use the resultant sort keys to put the entries in order.

```

2312 ITERATE {presort}
2313 SORT

```

If there are any duplicate author-year combinations, these functions detect them and disambiguate them by adding a letter to the year.

```

2314 STRINGS { last.label next.extra }
2315 INTEGERS { last.extra.num last.extra.num.extended last.extra.num.blank number.label }
2316 FUNCTION {initialize.extra.label.stuff}
2317 { #0 int.to.chr$ 'last.label :=
2318   "" 'next.extra :=
2319   #0 'last.extra.num :=

```

```

2320 "a" chr.to.int$ #1 - 'last.extra.num.blank :=
2321 last.extra.num.blank 'last.extra.num.extended :=
2322 #0 'number.label :=
2323 }
2324 FUNCTION {forward.pass}
2325 { last.label label =
2326   { last.extra.num #1 + 'last.extra.num :=
2327     last.extra.num "z" chr.to.int$ >
2328     { "a" chr.to.int$ 'last.extra.num :=
2329       last.extra.num.extended #1 + 'last.extra.num.extended :=
2330     }
2331     'skip$
2332     if$
2333     last.extra.num.extended last.extra.num.blank >
2334     { last.extra.num.extended int.to.chr$
2335       last.extra.num int.to.chr$
2336       * 'extra.label := }
2337     { last.extra.num int.to.chr$ 'extra.label := }
2338     if$
2339   }
2340   { "a" chr.to.int$ 'last.extra.num :=
2341     "" 'extra.label :=
2342     label 'last.label :=
2343   }
2344   if$
2345   number.label #1 + 'number.label :=
2346 }
2347 FUNCTION {reverse.pass}
2348 { next.extra "b" =
2349   { "a" 'extra.label := }
2350   'skip$
2351   if$
2352   extra.label 'next.extra :=
2353   extra.label
2354   duplicate$ empty$
2355   'skip$
2356   { "{\natexlab{" swap$ * "}} * }
2357   if$
2358   'extra.label :=
2359   label extra.label * 'label :=
2360 }

```

Applying these functions is a bit more tricky, but once we have done that, we re-sort the entries to make absolutely sure they in the right order, this time with the year between with `sort.label` and the title.

```

2361 EXECUTE {initialize.extra.label.stuff}
2362 ITERATE {forward.pass}
2363 REVERSE {reverse.pass}
2364 FUNCTION {bib.sort.order}
2365 { sort.label
2366   " "
2367   *
2368   year field.or.null sortify
2369   *
2370   " "
2371   *
2372   title field.or.null
2373   sort.format.title
2374   *
2375   #1 entry.max$ substring$
2376   'sort.key$ :=
2377 }
2378 ITERATE {bib.sort.order}
2379 SORT

```

We now write out the beginning of the bibliography to a .bbl file. This includes opening a \LaTeX environment, `thebibliography`, and some user-redefinable strings.

```

2380 FUNCTION {begin.bib}
2381 { preamble$ empty$
2382   'skip$
2383   { preamble$ write$ newline$ }

```



```

2384   if$
2385   "\begin{thebibliography}{\" number.label int.to.str$ * \"} *
2386   write$ newline$
2387   "\providecommand{\natexlab}[1]{#1}"
2388   write$ newline$
2389   "\providecommand{\url}[1]{\texttt{#1}}"
2390   write$ newline$
2391   "\providecommand{\urlprefix}{Available from: }"
2392   write$ newline$
2393   "\providecommand{\urldateprefix}{Accessed }"
2394   write$ newline$
2395   "\providecommand{\selectlanguage}[1]{\relax}"
2396   write$ newline$
2397   "\providecommand{\bibinfo}[2]{#2}"
2398   write$ newline$
2399   "\providecommand{\eprint}[2][\url{#2}]"
2400   write$ newline$
2401   }
2402 EXECUTE {begin.bib}
2403 EXECUTE {init.state.consts}

```

We go through each entry in turn, in the right order, and write the formatted text out to the file.

```

2404 ITERATE {call.type$}

```

After the last entry, we close the \LaTeX environment and stop writing to the .bib file.

```

2405 FUNCTION {end.bib}
2406 { newline$
2407   "\end{thebibliography}" write$ newline$
2408   }
2409 EXECUTE {end.bib}

```