

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

Maintainer: Alex Ball*

Package v7.1 – 26 January 2026

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

The files you need are included in the zip archive available from **GitHub**. If you use the zip archive from **CTAN**, you will need to run `luatex bath-bst.dtx` to generate them.

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 Managed way

The latest stable release of `bath-bst` has been packaged for TeX Live and MiKTeX. If you are running TeX Live and have `tlmgr` installed, you can install the package simply by running `tlmgr install bath-bst`. If you are running MiKTeX, you can install the package by running `mpm --install=bath-bst`. Both `tlmgr` and `mpm` have GUI versions that you might find friendlier.

1.1.2 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`. Ensure you have (at least) the `luatexja`, `adobemapping` and `ipaex` packages installed first.
- 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`.

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

1.1.3 Manual way

You do not need to follow the first step if you downloaded the zip archive from [GitHub](#). You do not need to follow the second step if you already have the PDF documentation.

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](#), [ipaex](#) and [haranoaji](#) 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.
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 BibTeX 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 BibTeX 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 introduced 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 BibTeX; 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`. From version 2, the advice is 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}
\setcitestyle{yysep={;}}
\setlength{\bibhang}{0pt}
\bibliographystyle{bathx}
```

In the unlikely event that you end up printing full references outside the reference list, you may need some extra command definitions:

```
\newcommand*{\urlprefix}{Available from: }
\newcommand*{\urldateprefix}{Accessed }
```

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

```
\bibliography{file}
```

3.1 Writing a citation

To make a citation in the text, use the key that corresponds to the entry in your .bib file. The following examples illustrate the principles laid out in the Harvard (Bath) guide.

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!

3.1.1 The essentials

Name occurring naturally

<code>\citet{blockley2021evs}</code> says\dots	Blockley (2021) says...
--	-------------------------

Name not occurring naturally

\dots this \citep{blockley2021evs}	...this (Blockley, 2021)
------------------------------------	--------------------------

Page information

<code>\citet[p.75]{burchard1965hhl}\</code> <code>\citep[para.4]{manco2013va}\</code> <code>\citep[00:28:41]{moran2016sol}</code>	Burchard (1965, p.75) (Manco, 2013, para.4) (Moran, 2016, 00:28:41)
---	---

Multiple sources in a single citation

<code>\citep{soper1972rbc, burrell1973ist, haurant2004bbh}\</code> <code>\citep{adams2009tc1, adams2014tc2, adams2017tc3}</code>	(Soper, 1972; Burrell, 1973; Haurant, 2004) (Adams, 2009; 2014; 2017)
---	--

Citing a document cited in another document

<code>\citep[Walters, 1883; cited by [p.87]{burrell1973ist}\</code>	(Walters, 1883; cited by Burrell, 1973, p.87)
---	---

3.1.2 General rules for authors

Corporate author

```
\citep{brbr1965gph}\  
\citep{gb.hc2003/04-30}\  
\citep{gb.pa2014}
```

(Bristol Region Building Record, 1965)
(Great Britain. Parliament. House of
Commons, 2004)
(*Pensions Act*, 2014)

Corporate author acronyms

```
World Health Organization  
\citep{who1986nde}
```

World Health Organization (WHO,
1986)

Multiple sources by same author(s) in same year

```
\citep{stieg1981cer} and  
\citep{stieg1981inh}
```

(Stieg, 1981a) and (Stieg, 1981b)

```
\citep{jonesNDa} and \citep{jonesNDb}
```

(Jones, n.d. a) and (Jones, n.d. b)

Different lead authors with the same surname

```
\citep{wang2018tc} and  
\citep{wang2020tc}
```

(Wang, J., 2018) and (Wang, K., 2020)

3.1.3 Multiple authors

Two authors

```
\citep{shah.corrick2016hsc}
```

(Shah and Corrick, 2016)

Three authors

```
\citep{devlin.etal2021ipp}
```

(Devlin, Martin and Ostrovnaya, 2021)

Four or more authors

```
\citep{solomon.etal2019cbe}
```

(Solomon et al., 2019)

Groups of authors where different lead authors have the same surname

```
\citep{wang.etal2002tc6} and  
\citep{wang.etal2002tc7}
```

(Wang, J. et al., 2002) and (Wang, K.
et al., 2002)

Different groups of authors with the same lead author

```
\citep{smith.etal2020tc8} and  
\citep{smith.etal2020tc9}
```

(Smith, Taylor et al., 2020) and (Smith,
Jones et al., 2020)

4 Breaking changes between versions

The following sections list implementation changes that might affect the way in which you write your bibliography database files.

In addition to the changes listed, when upgrading between major versions, you should expect some changes in formatting relating to the evolution of the Harvard (Bath) style itself. Some of the style's internal settings and structures may also change without warning.

4.1 Version 3

- Journal titles are now coerced to sentence case, so any capital letters you want to keep need to be protected with braces.
- Entries of type **unpublished** are now marked as unpublished. If you have been using this for an item that should not be marked as unpublished (e.g. archive photographs, law reports), you can use **booklet** instead: this is the other entry type (along with **misc**) that has a **howpublished** field instead of a **publisher**.
- Some changes to the Harvard (Bath) style cannot be applied automatically (e.g. the update to how standards are referenced), so you may need to update your **.bib** file accordingly.



4.2 Version 5

- In the 2021 revision of the style, the titles of broadcast media (TV, radio) became upright, with series information for titled episodes being italic. To get this new formatting, you must use the **type** field as shown.
- Series names (except those in **legislation** and **jurisdiction** entries) are now coerced to sentence case, so any capital letters you want to keep need to be protected with braces.
- The '[Online]' tag is now also triggered by **urldate/urlyear**. In cases where it occurs directly after the title, it is no longer left behind (after the year) if the title is moved to the head of the reference.
- The **nameaddon** field is now supported for annotating author lists with, say, a social media handle.
- The **article** and **manual** entry types now support the **month** field.
- Maps need to be given the **book** entry type; previously **manual** was recommended.

4.3 Version 6

- Entry type clarifications such as 'Photograph' and 'Computer program' are now handled differently from translated titles, in particular as regards their position relative to the version, and how they interact with the '[Online]' tag. As a result, they should now be placed in **entrysubtype** rather than **titleaddon**.
- All elements of the title block (including **titleaddon**, **version**, and **entrysubtype**) now move to the head of the reference in the absence of an author/editor.



5 Examples

The examples below are shown in three parts. The first, marked with , shows an extract from the *Referencing guide: Harvard Bath* 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: 'RX' indicates the 'Reference examples (A-Z)' section of the Guide; 'WC' indicates the 'Write a citation' section of the Guide; 'RL' indicates the '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.



5.1 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}}
```

RX

-  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}}
```



RX

-  Solomon, M.R., Askegaard, S., Hogg, M. and Bamossy, G.J., 2019. *Consumer behaviour: a European perspective*. 7th ed. Harlow: Pearson.
-  Solomon, M.R., Askegaard, S., Hogg, M. and Bamossy, G.J., 2019. *Consumer behaviour: a European perspective*. 7th ed. Harlow: Pearson.

```
@book{solomon.etal2019cbe,  
  author = {Solomon, M. R. and Askegaard, S. and Hogg, M. and Bamossy, G. J.},  
  year = {2019},  
  title = {Consumer Behaviour: a {European} Perspective},  
  edition = {7},  
  address = {Harlow},  
  publisher = {Pearson}}
```


RX

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}}
```

RX

 Internally, collection is an alias for book.

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}}
```

RX


 Blockley, D., 2021. *Engineering: a very short introduction* [Online]. Oxford: Oxford University Press. Available from: <https://www.amazon.co.uk/kindle> [Accessed 30 June 2021].

 Blockley, D., 2021. *Engineering: a very short introduction* [Online]. Oxford: Oxford University Press. Available from: <https://www.amazon.co.uk/kindle> [Accessed 30 June 2021].


```
@book{blockley2021evs,
  author = {Blockley, D.},
  year = {2021},
  title = {Engineering: a Very Short Introduction},
  address = {Oxford},
  publisher = {Oxford University Press},
  url = {https://www.amazon.co.uk/kindle},
  urlyear = {30 June 2021}}
```

RX

Book known by its title


-  Unfortunately, to avoid breaking other use cases for the **book** entry type, to achieve the following format you must use the non-standard **reference** entry type.

 British National Formulary, 2020. 79th ed. London: Pharmaceutical Press.

 British National Formulary, 2020b. 79th ed. London: Pharmaceutical Press.

```
@reference{bnf2020,
  year = {2020},
  title = {{British National Formulary}},
  edition = {79},
  address = {London},
  publisher = {Pharmaceutical Press}}
```

RX

-  Similarly, to achieve the following formats you must use the non-standard **inreference** entry type.

- 📖 British National Formulary, 2020. 79th ed. *Aspirin*. London: Pharmaceutical Press, pp.280–281.
- ⚙ British National Formulary, 2020a. 79th ed. *Aspirin*. London: Pharmaceutical Press, pp.280–281.

```
@inreference{aspirin2020bnf,
  title = {Aspirin},
  year = {2020},
  booktitle = {{British National Formulary}},
  edition = {79},
  address = {London},
  publisher = {Pharmaceutical Press},
  pages = {280–281}}
```

RX

- 📖 British National Formulary, 2019. *Aspirin* [Online]. London: Pharmaceutical Press. Available from: https://www.medicinescomplete.com/#/content/bnf/_456850132 [Accessed 26 November 2019].
- ⚙ British National Formulary, 2019. *Aspirin* [Online]. London: Pharmaceutical Press. Available from: https://www.medicinescomplete.com/#/content/bnf/_456850132 [Accessed 26 November 2019].

```
@inreference{aspirin2019bnf,
  title = {Aspirin},
  year = {2019},
  booktitle = {{British National Formulary}},
  address = {London},
  publisher = {Pharmaceutical Press},
  url = {https://www.medicinescomplete.com/#/content/bnf/_456850132},
  urlyear = {26 November 2019}}
```


RX


- ❗ If you are using the same .bib file with Bib_T_EX and bib_lat_ex, one unfortunate difference is that you must escape the hash symbol inside URLs with Bib_T_EX but you must not escape it inside a URL with bib_lat_ex.
- ❗ Internally, `inreference` is an alias for `incollection`. Use `inreference` if you need compatibility with bib_lat_ex-bath.
- ❗ The style ensures the `booktitle` is used in title citation commands.

```
\cite{aspirin2019bnf}, \citet{bnf2020}, \citet{aspirin2020bnf}.
```

British National Formulary (2019), British National Formulary (2020b), British National Formulary (2020a).


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}}
```

RL

 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}}
```

RX

5.2 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}}
```

RL

- 📖 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}}
```

RL

- 📖 Newman, R., 2010. Malaria control beyond 2010. *British medical journal*, 341(7765), pp.157–208.
- ⚙️ Newman, R., 2010. Malaria control beyond 2010. *British medical journal*, 341(7765), pp.157–208.

```
@article{newman2010mcb,
  author = {Newman, R.},
  year = {2010},
  title = {Malaria control beyond 2010},
  journal = {{British} Medical Journal},
  volume = {341},
  number = {7765},
  pages = {157-208}}
```

RX

- 📖 Wetzstein, G., Ozcan, A., Gigan, S., Fan, S., Englund, D., Soljacic, M., Denz, C., Miller, D.A.B. and Psaltis, D., 2020. Inference in artificial intelligence with deep optics and photonics. *Nature*, 588(7836), pp.39–47.
- ⚙️ Wetzstein, G., Ozcan, A., Gigan, S., Fan, S., Englund, D., Soljacic, M., Denz, C., Miller, D.A.B. and Psaltis, D., 2020. Inference in artificial intelligence with deep optics and photonics. *Nature*, 588(7836), pp.39–47.

```
@article{wetzstein.etal2020iai,
  author = {Wetzstein, G. and Ozcan, A. and Gigan, S. and Fan, S. and
    Englund, D. and Soljacic, M. and Denz, C. and Miller, D. A. B. and
    Psaltis, D.},
  year = {2020},
  title = {Inference in Artificial Intelligence with Deep Optics and Photonics},
  journal = {Nature},
  volume = {588},
  number = {7836},
  pages = {39-47}}
```

RX

Electronic journal article

📄 Devlin, S.M., Martin, A. and Ostrovnaya, I., 2021. Identifying prognostic pairwise relationships among bacterial species in microbiome studies. *PLOS computational biology* [Online], 17(11). Available from: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009501> [Accessed 9 December 2021].

⚙️ Devlin, S.M., Martin, A. and Ostrovnaya, I., 2021. Identifying prognostic pairwise relationships among bacterial species in microbiome studies. *PLOS computational biology* [Online], 17(11). Available from: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009501> [Accessed 9 December 2021].

```
@article{devlin.etal2021ipp,  
  author = {Devlin, S. M. and Martin, A. and Ostrovnaya, I.},  
  year = {2021},  
  title = {Identifying Prognostic Pairwise Relationships Among Bacterial  
    Species in Microbiome Studies},  
  journal = {{PLOS} Computational Biology},  
  volume = {17},  
  number = {11},  
  url = {https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009501},  
  urlyear = {9 December 2021}}
```

RX

📄 Steward, S., Connelly, D. and Robinson, J., 2020. Everything you should know about the coronavirus outbreak. *The pharmaceutical journal* [Online]. Available from: <https://www.pharmaceutical-journal.com/news-and-analysis/features/everything-you-should-know-about-the-coronavirus-outbreak/20207629.article> [Accessed 30 April 2020].

⚙️ Steward, S., Connelly, D. and Robinson, J., 2020. Everything you should know about the coronavirus outbreak. *The pharmaceutical journal* [Online]. Available from: <https://www.pharmaceutical-journal.com/news-and-analysis/features/everything-you-should-know-about-the-coronavirus-outbreak/20207629.article> [Accessed 30 April 2020].

```
@article{steward.etal2020eys,  
  author = {Steward, S. and Connelly, D. and Robinson, J.},  
  year = {2020},  
  title = {Everything You Should Know About the Coronavirus Outbreak},  
  journal = {The Pharmaceutical Journal},  
  url = {https://www.pharmaceutical-journal.com/news-and-analysis/features/everything-you-should-know-about-the-coronavirus-outbreak/20207629.article},  
  urlyear = {30 April 2020}}
```

RX

- 💡 If citing an article yet to be officially published, use the `pubstate` field (from `biblatex`) with the keyword `inpress` for ‘in press’ and `inpreparation` or `submitted` (whichever is more accurate) for ‘preprint’.

- 📖 Liontou, C., Kontopodis, E., Oikonomidis, N., Maniotis, C., Tassopoulos, A., Tsiafoutis, I., Lazaris, E. and Koutouzis, M., 2019. Distal radial access: a review article. *Cardiovascular revascularization medicine* [Online], in press. Available from: <https://www.sciencedirect.com/science/article/pii/S1553838919303367> [Accessed 19 June 2019].
- ⚙️ Liontou, C., Kontopodis, E., Oikonomidis, N., Maniotis, C., Tassopoulos, A., Tsiafoutis, I., Lazaris, E. and Koutouzis, M., 2019. Distal radial access: a review article. *Cardiovascular revascularization medicine* [Online], in press. Available from: <https://www.sciencedirect.com/science/article/pii/S1553838919303367> [Accessed 19 June 2019].

```
@article{liontou.etal2019dra,
  author = {Liontou, C. and Kontopodis, E. and Oikonomidis, N. and Maniotis, C. and
    Tassopoulos, A. and Tsiafoutis, I. and Lazaris, E. and Koutouzis, M.},
  year = {2019},
  title = {Distal Radial Access: a Review Article},
  journal = {Cardiovascular Revascularization Medicine},
  pubstate = {inpress},
  url = {https://www.sciencedirect.com/science/article/pii/S1553838919303367},
  urlyear = {19 June 2019}}
```

RX

Preprint or trial study 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].

```
@report{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}}
```

RX

- 💡 The Harvard (Bath) style treats preprints as grey literature, and so the `techreport` entry type (or its alias `report`) is the best match semantically.
- 💡 Use `library` to specify the name of the preprint repository. Note that the workaround in `bath.bst` of using `note` instead does not work in `bathx.bst`.

- 📖 Ganju, V., 2021. *A study of EnGeneIC Dream Vectors (EDV's) packaged with the chemotherapy, E-EDV-D682 given simultaneously as non-targeted EDVs carrying an immune enhancer called EDV-GC, in participants with advanced pancreatic and other cancers whose disease has progressed after one or two treatment regimes, or where other standard therapies are not appropriate.* Sydney: University of Sydney. *Australian and New Zealand Clinical Trials Registry* [Online]. Available from: <http://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=365258> [Accessed 29 April 2021].
- ⚙ Ganju, V., 2021. *A study of EnGeneIC Dream Vectors (EDV's) packaged with the chemotherapy, E-EDV-D682 given simultaneously as non-targeted EDVs carrying an immune enhancer called EDV-GC, in participants with advanced pancreatic and other cancers whose disease has progressed after one or two treatment regimes, or where other standard therapies are not appropriate.* Sydney: University of Sydney. *Australian and New Zealand Clinical Trials Registry* [Online]. Available from: <http://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=365258> [Accessed 29 April 2021].

```
@report{ganju2021sed,
  author = {Ganju, V.},
  year = {2021},
  title = {A study of {EnGeneIC} {Dream} {Vectors} {(EDV's)} packaged with the chemotherapy,
    {E-EDV-D682} given simultaneously as non-targeted {EDVs} carrying an immune enhancer
    called {EDV-GC}, in participants with advanced pancreatic and other cancers whose disease
    has progressed after one or two treatment regimes, or where other standard therapies are
    not appropriate},
  address = {Sydney},
  institution = {University of Sydney},
  library = {Australian and New Zealand Clinical Trials Registry},
  url = {http://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=365258},
  urlyear = {29 April 2021}}
```

RX

Newspaper article

- 🔑 Give the issue's day and month as the month or 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}},
  month = {{29 July}},
  pages = {16c}}
```

RX

- ❗ The extra braces around the month field aren't necessary with Bib_TE_X, but are needed if you want compatibility with biblatex.

- 📖 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}},
  month = {{4 June}},
  pages = {28a}}
```

RX

📄 Cogley, M., 2020. Corporate confidence slumps to record low. *The Telegraph* [Online], 4 May. Available from: <https://www.telegraph.co.uk/business/2020/05/03/corporate-confidence-slumps-all-time-low/> [Accessed 5 May 2020].

⚙️ Cogley, M., 2020. Corporate confidence slumps to record low. *The Telegraph* [Online], 4 May. Available from: <https://www.telegraph.co.uk/business/2020/05/03/corporate-confidence-slumps-all-time-low/> [Accessed 5 May 2020].

```
@article{cogley2020ccs,
  author = {Cogley, M.},
  year = {2020},
  title = {Corporate Confidence Slumps to Record Low},
  journal = {{The Telegraph}},
  volume = {4 May},
  url = {https://www.telegraph.co.uk/business/2020/05/03/corporate-confidence-slumps-all-time-low/},
  urlyear = {5 May 2020}}
```

RX

5.3 Conference papers

- ❗ As in the standard BibT_EX 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}}
```

RX

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}}
```

RX

5.4 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}}
```

RX

- ❗ Internally, `thesis` is an alias for `phdthesis`. If you need compatibility with `biblatex-bath` and don't want to display a thesis type, use `thesis`.

Report, working paper, or research briefing

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

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

RX

- ☞ BRE, 2007. *Designing quality buildings: a BRE guide*. (Report 497). Bracknell: BRE.
- ⚙ BRE, 2007. *Designing quality buildings: a BRE guide*. (Report 497). Bracknell: BRE.

```
@techreport{bre2007dqb,
  author = {{BRE}},
  year = {2007},
  title = {Designing Quality Buildings: a {BRE} Guide},
  address = {Bracknell},
  institution = {BRE},
  type = {Report},
  number = {497}}
```

RX

- ☞ Deneulin, S. and Dinerstein, A.C., 2010. *Hope movements: social movements in the pursuit of human development*. (Bath papers in international development and wellbeing, no. 8). Bath: University of Bath.
- ⚙ Deneulin, S. and Dinerstein, A.C., 2010. *Hope movements: Social movements in the pursuit of human development*. (Bath papers in international development and wellbeing, no. 8). Bath: University of Bath.

```
@report{deneulin.dinerstein2010hms,
  author = {Deneulin, S. and Dinerstein, A. C.},
  year = {2010},
  title = {Hope Movements: Social Movements in the Pursuit of Human Development},
  address = {Bath},
  institution = {University of Bath},
  series = {Bath Papers in International Development and Wellbeing},
  number = {no.~8}}
```

RX

- ❗ Internally, **report** is an alias for **techreport**. If you need compatibility with biblatex-bath and don't want to display a report type, use **report**.
- ❗ The style inserts a comma between the **series** and **number** fields, but does not insert one between **type** and **number**.



- ☞ WHO (World Health Organization), 1986. *Nutrition and development in East Africa*. Geneva: WHO.
- ⚙ WHO (World Health Organization), 1986. *Nutrition and development in East Africa*. Geneva: WHO.

```
@report{who1986nde,
  author = {{World Health Organization}},
  shortauthor = {WHO},
  title = {Nutrition and Development in {East Africa}},
  year = {1986},
  address = {Geneva},
  institution = {WHO}}
```

WC



- 💡 To abbreviate any of the names in the **author** field (or **editor**, if that appears at the head of the reference), provide a **shortauthor** (**shorteditor**) field. This should contain the same set of names as the **author** (**editor**) field: replace long names with their abbreviations and leave the others as they are.

Standard

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


```
@standard{bs5605:1990,  
  author = {{BSI}},  
  year = {1990},  
  title = {{BS}-5605:1990 {Recommendations} for Citing and Referencing Published Material},  
  address = {London},  
  organization = {BSI}}
```

RX



-  ASTM, 2019. *ASTM D1655 - 19 Standard specification for aviation turbine fuels*. West Conshohocken, Pa.: ASTM.
-  ASTM, 2019. *ASTM D1655 - 19 Standard specification for aviation turbine fuels*. West Conshohocken, Pa.: ASTM.

```
@standard{astm.d1655,  
  author = {{ASTM}},  
  year = {2019},  
  title = {{ASTM-D1655} - 19 {Standard} Specification for Aviation Turbine Fuels},  
  address = {West Conshohocken, Pa.},  
  organization = {ASTM}}
```

RX


 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}}
```

RX

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

Unpublished written material and personal communications

-  Harris, G., 2013. *Focus group recommendations: internal task group report*. Unpublished.
-  Harris, G., 2013. *Focus group recommendations: internal task group report*. Unpublished.

```
@unpublished{harris2013fgr,  
  author = {Harris, G.},  
  year = {2013},  
  title = {Focus group recommendations: internal task group report}}
```

RX

- 📖 Hadley, S., 2015. *Biomechanics: introductory reading, BM289: sport biomechanics*. University of Bath. Unpublished.
- ⚙️ Hadley, S., 2015. *Biomechanics: introductory reading, BM289: sport biomechanics*. University of Bath. Unpublished.

```
@unpublished{hadley2015bir,
  author = {Hadley, S.},
  year = {2015},
  title = {Biomechanics: introductory reading, {BM289}: sport biomechanics},
  howpublished = {University of Bath}}
```

RX

- 📖 Thomas, D., 2015. Word count and referencing style. *Frequently asked questions discussion board: PHYS 2011: housing studies*. University of Bath. Unpublished.
- ⚙️ Thomas, D., 2015. Word count and referencing style. *Frequently asked questions discussion board: PHYS 2011: housing studies*. University of Bath. Unpublished.

```
@unpublished{thomas2015wcr,
  author = {Thomas, D.},
  year = {2015},
  title = {Word count and referencing style},
  booktitle = {Frequently asked questions discussion board: {PHYS} 2011: housing studies},
  howpublished = {University of Bath}}
```

RX

5.5 Audiovisual materials

Image

- 📖 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].
- ⚙️ 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.

📖 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].

⚙ 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

📖 Bristol Region Building Record, 1965. *Green Park House (since demolished), viewed from southwest* [Photograph]. BRBR, D/877/1. Archives & Research Collections, University of Bath Library.

⚙ Bristol Region Building Record, 1965. *Green Park House (since demolished), viewed from southwest* [Photograph]. BRBR, D/877/1. Archives & Research Collections, University of Bath Library.

```
@image{brbr1965gph,
  author = {{Bristol Region Building Record}},
  year = {1965},
  title = {{Green} {Park} {House} (since demolished), viewed from southwest},
  titleaddon = {Photograph},
  number = {BRBR, D/877/1},
  library = {Archives & Research Collections, University of Bath Library}}
```

RI

💡 Use the **library** field to record the archive. If the image is on display rather than archived, use **institution** instead for the museum, gallery or building. 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.

💡 Put archival locator information (e.g. box or shelf number) in the **number** field.

💡 Put the clarification of the resource type in **entrysubtype**. In most cases, you can use **titleaddon** instead, but **entrysubtype** will correctly suppress 'Online' being shown while **titleaddon** will not.

❗ The **@image** entry type can be used for any visual artistic work, including sculptures and interactive exhibits.

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.

```
@book{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}}
```

RX

📖 Ordnance Survey, 2020. *Street view map of University of Bath* [Online], 1:5000, OS VectorMap® Local. Available from: <https://digimap.edina.ac.uk/roam/map/os> [Accessed 30 April 2020].

⚙️ Ordnance Survey, 2020. *Street view map of University of Bath* [Online], 1:5000, OS VectorMap® Local. Available from: <https://digimap.edina.ac.uk/roam/map/os> [Accessed 30 April 2020].

```
@book{os2020bath,  
  author = {{Ordnance Survey}},  
  year = {2020},  
  title = {Street view map of {University of Bath}},  
  series = {1:5000, {OS VectorMap® Local}},  
  url = {https://digimap.edina.ac.uk/roam/map/os},  
  urlyear = {30 April 2020}}
```

RX

📖 Google, 2020. *Harbourside, Bristol* [Online], Google Maps. Available from: <https://www.google.co.uk/maps/place/Harbourside,+Bristol/> [Accessed 30 April 2020].

⚙️ Google, 2020. *Harbourside, Bristol* [Online], Google Maps. Available from: <https://www.google.co.uk/maps/place/Harbourside,+Bristol/> [Accessed 30 April 2020].

```
@book{google2020harbourside,  
  author = {{Google}},  
  year = {2020},  
  title = {{Harbourside, Bristol}},  
  series = {{Google Maps}},  
  url = {https://www.google.co.uk/maps/place/Harbourside,+Bristol/},  
  urlyear = {30 April 2020}}
```

RX

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}}
```

RX

❗ Internally, movie and video are aliases for booklet.

📖 *The elephant man* [Online], 1980. Film. Directed by David Lynch. USA: Brooksfilms. Available from: BBC iPlayer [Accessed 4 May 2021].

⚙️ *The elephant man* [Online], 1980. Film. Directed by David Lynch. USA: Brooksfilms. Available from: BBC iPlayer [Accessed 4 May 2021].

```
@video{elephant-man1980,  
  year = {1980},  
  title = {The elephant man},  
  type = {Film},  
  note = {Directed by David Lynch},  
  address = {USA},  
  publisher = {Brooksfilms},  
  library = {BBC iPlayer},  
  urlyear = {4 May 2021}}
```

RX

Online video/audio

📺 Moran, C., 2016. *Save our libraries* [Online]. Available from: <https://youtu.be/gKTfCz4JtVE> [Accessed 29 April 2016].

⚙️ Moran, C., 2016. *Save our libraries* [Online]. Available from: <https://youtu.be/gKTfCz4JtVE> [Accessed 29 April 2016].

```
@video{moran2016sol,  
  author = {Moran, C.},  
  year = {2016},  
  title = {Save Our Libraries},  
  url = {https://youtu.be/gKTfCz4JtVE},  
  urlyear = {29 April 2016}}
```

RX

📺 Chakrabarti, V., 2016. *How architecture and city planning can combat social inequality* [Online]. Available from: <https://www.curbed.com/2016/5/5/11593058/vishaan-chakrabarti-pau-curbed-appeal-podcast> [Accessed 28 March 2019].

⚙️ Chakrabarti, V., 2016. *How architecture and city planning can combat social inequality* [Online]. Available from: <https://www.curbed.com/2016/5/5/11593058/vishaan-chakrabarti-pau-curbed-appeal-podcast> [Accessed 28 March 2019].

```
@audio{chakrabarti2016hac,  
  author = {Chakrabarti, V.},  
  year = {2016},  
  title = {How Architecture and City Planning Can Combat Social Inequality},  
  url = {https://www.curbed.com/2016/5/5/11593058/vishaan-chakrabarti-pau-curbed-appeal-podcast},  
  urlyear = {28 March 2019}}
```

RI

Television or radio broadcast

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

💡 If the episode has an individual title, use **series** for the series and **number** for the episode number. Otherwise, give both series and episode number as the **title**.

📺 Hurry up and wait, 2021. *Inside no.9, Episode 6:4*. TV. BBC2, 31 May. 21.30 hrs.

⚙️ Hurry up and wait, 2021. *Inside no.9, Episode 6:4*. TV. BBC2, 31 May. 21.30 hrs.

```
@video{inn2021huw,  
  year = {2021},  
  title = {Hurry up and wait},  
  series = {Inside No.9},  
  number = {{Episode} 6:4},  
  type = {TV},  
  note = {BBC2, 31 May. 21.30 hrs}}
```

RX

📺 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}}
```

RX

- 📖 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}}
```

RX

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

- 📖 The secret, 2020. *Tiger king: murder, mayhem and madness, Episode 3* [Online]. TV. Available from: Netflix [Accessed 4 May 2021].
- ⚙ The secret, 2020. *Tiger king: Murder, mayhem and madness, Episode 3* [Online]. TV. Available from: Netflix [Accessed 4 May 2021].

```
@video{tkmmm2020ts,
  year = {2020},
  title = {The Secret},
  series = {Tiger King: Murder, Mayhem and Madness},
  number = {{Episode} 3},
  type = {TV},
  library = {Netflix},
  urlyear = {4 May 2021}}
```

RX

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}}
```

RX

5.6 Digital media

Website or 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`.

- 📖 World Health Organization, 2018. *The top 10 causes of death* [Online]. Geneva: World Health Organization. Available from: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death> [Accessed 29 June 2020].
- ⚙ World Health Organization, 2018. *The top 10 causes of death* [Online]. Geneva: World Health Organization. Available from: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death> [Accessed 29 June 2020].

```
@online{who2018ttc,
  author = {{World Health Organization}},
  year = {2018},
  title = {The Top 10 Causes of Death},
  address = {Geneva},
  organization = {World Health Organization},
  url = {https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death},
  urlyear = {29 June 2020}}
```

RX

- 📖 Manco, J., 2013. *Vernacular architecture* [Online]. Available from: <https://www.buildinghistory.org/style/vernacular.shtml> [Accessed 20 February 2020].
- ⚙️ Manco, J., 2013. *Vernacular architecture* [Online]. Available from: <https://www.buildinghistory.org/style/vernacular.shtml> [Accessed 20 February 2020].

```
@online{manco2013va,
  author = {Manco, J.},
  year = {2013},
  title = {Vernacular Architecture},
  url = {https://www.buildinghistory.org/style/vernacular.shtml},
  urlyear = {20 February 2020}}
```

RX

This template is also used for retrievable Generative AI content.

- 📖 ChatGPT, 2023. *Why is citing and referencing your sources important?* [Online]. San Francisco, Calif.: OpenAI. Available from: <https://chat.openai.com/share/782cb099-a0dc-45b1-8da4-0e99713f2d45> [Accessed 4 September 2023].
- ⚙️ ChatGPT, 2023. *Why is citing and referencing your sources important?* [Online]. San Francisco, Calif.: OpenAI. Available from: <https://chat.openai.com/share/782cb099-a0dc-45b1-8da4-0e99713f2d45> [Accessed 4 September 2023].

```
@online{chatgpt2023wcr,
  author = {{ChatGPT}},
  year = {2023},
  title = {Why is Citing and Referencing Your Sources Important?},
  address = {San Francisco, Calif.},
  organization = {OpenAI},
  url = {https://chat.openai.com/share/782cb099-a0dc-45b1-8da4-0e99713f2d45},
  urlyear = {4 September 2023}}
```

RX

Social media

- 📖 Gaiman, N., 2021. *Facebook post* [Online], 21 April. Available from: <https://www.facebook.com/neilgaiman/posts/305187897642814> [Accessed 13 May 2021].
- ⚙️ Gaiman, N., 2021. *Facebook post* [Online], 21 April. Available from: <https://www.facebook.com/neilgaiman/posts/305187897642814> [Accessed 13 May 2021].

```
@online{gaiman2021fbp,
  author = {Gaiman, N.},
  year = {2021},
  title = {Facebook post},
  month = {{21 April}},
  url = {https://www.facebook.com/neilgaiman/posts/305187897642814},
  urlyear = {13 May 2021}}
```

RX

📖 Library at University of Bath [@bathunilib], 2021. *Twitter post* [Online], 10 May. Available from: <https://twitter.com/BathUniLibrary/status/1391774402618998795?s=20> [Accessed 12 May 2021].

⚙️ Library at University of Bath [@bathunilib], 2021. *Twitter post* [Online], 10 May. Available from: <https://twitter.com/BathUniLibrary/status/1391774402618998795?s=20> [Accessed 12 May 2021].

```
@online{ublib2021tp,
  author = {{Library at University of Bath}},
  nameaddon = {@bathunilib},
  year = {2021},
  title = {Twitter post},
  month = {{10 May}},
  url = {https://twitter.com/BathUniLibrary/status/1391774402618998795?s=20},
  urlyear = {12 May 2021}}
```

RX

📖 University of Bath Library [@unibathlib], 2021. *Instagram post* [Online], 15 March. Available from: https://www.instagram.com/p/CMb5GHjFv9A/?utm_source=ig_web_copy_link [Accessed 7 May 2021].

⚙️ University of Bath Library [@unibathlib], 2021. *Instagram post* [Online], 15 March. Available from: https://www.instagram.com/p/CMb5GHjFv9A/?utm_source=ig_web_copy_link [Accessed 7 May 2021].

```
@online{ublib2021ip,
  author = {{University of Bath Library}},
  nameaddon = {@unibathlib},
  year = {2021},
  title = {Instagram post},
  month = {{15 March}},
  url = {https://www.instagram.com/p/CMb5GHjFv9A/?utm_source=ig_web_copy_link},
  urlyear = {7 May 2021}}
```

RX

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}}
```

RX

💡 Use the `journal` field to specify the mailing list. Internally, a `letter` entry with a `journal` field is treated like an `article` entry, while one without is treated like 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)


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].


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

RX

 Use the `title` field for the entry title, and the `library` field for the name of the database itself.


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* [Online]. 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},  
  publisher = {University of Bath},  
  doi = {10.15125/BATH-00069},  
  urlyear = {20 April 2016}}
```

RX

 You can use `dataset` instead of `online` as 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},  
  entrysubtype = {computer program},  
  url = {https://screencast-o-matic.com/},  
  urlyear = {16 May 2016}}
```

RX

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

 Put the clarification of the resource type in `entrysubtype`. In most cases, you can use `titleaddon` instead, but `titleaddon` is shown immediately after the title and may be shown

alongside the ‘Online’ label, whereas `entrysubtype` is printed after the `version` and supersedes the ‘Online’ label.

5.7 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}}
```

RX

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}}
```

RX

💡 Use the `titleaddon` field to supply the English translation of the title.

- 📖 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}}
```

RX

Work in a non-Roman alphabet

- 📖 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 \emph{中國社會科學}},
  volume = {1},
  pages = {168-79}}
```

RX

- 💡 The above 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. The trick is not portable to `biblatex-bath`.

- 💡 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}}
```

RX

- 📖 Pamporov, A., 2006. *Romskoto vsekidnevie v Balgariya* [Roma everyday life in Bulgaria]. Veliko Tarnovo: Faber.
- ⚙️ Pamporov, A., 2006. *Romskoto vsekidnevie v Balgariya* [Roma everyday life in Bulgaria]. Veliko Tarnovo: Faber.

```
@book{pamporov2006rvb,
  author = {Pamporov, A.},
  year = {2006},
  title = {Romskoto vsekidnevie v {Balgariya}},
  titleaddon = {Roma everyday life in Bulgaria},
  address = {Veliko Tarnovo},
  publisher = {Faber}}
```

RX

5.8 Government and parliamentary document

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

```
@report{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}}
```

RX

💡 To achieve compatibility with biblatex-bath, you can either set the **type** field to the empty string or use the **report** entry type, which is an alias for **techreport**.

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*. (HL 1986/87, (66)). London: HMSO.
- ⚙️ Great Britain. Parliament. House of Lords, 1987. *Social fund (maternity and funeral expenses) bill*. (HL 1986/87, (66)). London: HMSO.

```
@report{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}}
```

RX

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

```
@report{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}}
```

RX

- 💡 To support the Command Paper template, if you give a report the **series** ‘C’, ‘Cd’, ‘Cmd’, ‘Cmnd’ or ‘Cm’, it is followed by a dot rather than a comma. (If you need the comma as normal, wrap the value in an extra pair of braces.)

If you prefer, you can give the Command Paper abbreviation in **type** and supply the dot yourself.

House of Commons/House of Lords bill

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

```
@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}}
```

RX

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}}
```

RX

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}}
```

RX

Statutory instrument

- 📖 *The Human Medicines Regulations 2012* [Online], No.1916, United Kingdom: HMSO. Available from: <http://www.legislation.gov.uk/uksi/2012/1916/contents> [Accessed 22 April 2021].
- ⚙️ *The Human Medicines Regulations 2012* [Online], No.1916, United Kingdom: HMSO. Available from: <http://www.legislation.gov.uk/uksi/2012/1916/contents> [Accessed 22 April 2021].

```
@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/contents},
  urlyear = {22 April 2021}}
```

RX

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

Parliamentary debate

- 📖 Great Britain. Parliament. House of Commons, 2024. Rwanda Plan Cost and Asylum System. *Hansard* [Online], 744, 9 January 2024. Available from: <https://hansard.parliament.uk/Commons/2024-01-09/debates/57105D77-72DE-4246-9F6E-ED87FCB9A088/RwandaPlanCostAndAsylumSystem> [Accessed 12 December 2024].
- ⚙️ Great Britain. Parliament. House of Commons, 2024. Rwanda Plan Cost and Asylum System. *Hansard* [Online], 744, 9 January 2024. Available from: <https://hansard.parliament.uk/Commons/2024-01-09/debates/57105D77-72DE-4246-9F6E-ED87FCB9A088/RwandaPlanCostAndAsylumSystem> [Accessed 12 December 2024].

```
@legal{gb.hc2024rpc,
  entrysubtype = {parliamentary},
  author = {{Great Britain. Parliament. House of Commons}},
  year = {2024},
  title = {{Rwanda Plan Cost and Asylum System}},
  journal = {Hansard},
  volume = {744},
  eventyear = {9 January 2024},
  url = {https://hansard.parliament.uk/Commons/2024-01-09/debates/57105D77-72DE-4246-9F6E-ED87FCB9A088/RwandaPlanCostAndAsylumSystem},
  urlyear = {12 December 2024}}
```

RX

5.9 Legal document: EU reports/legislation

EU publication

The following example has been removed.

- 📖 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.

```
@report{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}}
```

RX

EU regulation or directive, decision, recommendation or opinion

- 📖 Council Regulation (EC) 1984/2003 of 8 April 2003 introducing a system for the statistical monitoring of trade in bluefin tuna, swordfish and big eye tuna within the Community [2003] *OJ* L295.
- ⚙️ Council Regulation (EC) 1984/2003 of 8 April 2003 introducing a system for the statistical monitoring of trade in bluefin tuna, swordfish and big eye tuna within the Community [2003] *OJ* L295.

```
@legislation{eu.dir1984/2003,
  title = {Council {Regulation} ({EC}) 1984/2003 of 8 {April} 2003 Introducing a System for
    the Statistical Monitoring of Trade in Bluefin Tuna, Swordfish and Big Eye Tuna within
    the {Community}},
  year = {2003},
  journal = {OJ},
  series = {L},
  volume = {295}}
```

RX

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

<pre>\defcitealias{eu.dir1984/2003}{% Council Regulation [EC] 1984/2003} \citepalias{eu.dir1984/2003}</pre>	<p>(Council Regulation [EC] 1984/2003)</p>
---	--

5.10 Legal document: case reports

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}}
```

RX

- 💡 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`.

Legal case study

📖 *Seldon v Clarkson Wright & Jakes*, 2009. England and Wales Court of Appeal (Civil Division), 889. *Bailii* [Online]. Available from: <http://www.bailii.org/ew/cases/EWCA/Civ/2009/889.html> [Accessed 23 April 2024].

⚙️ *Seldon v Clarkson Wright & Jakes*, 2009. England and Wales Court of Appeal (Civil Division), 889. *Bailii* [Online]. Available from: <http://www.bailii.org/ew/cases/EWCA/Civ/2009/889.html> [Accessed 23 April 2024].

```
@jurisdiction{seldon-v-c.w.j2009,
  title = {Seldon v-{Clarkson} {Wright} \& {Jakes}},
  year = {2009},
  institution = {England and Wales Court of Appeal (Civil Division)},
  number = {889},
  journal = {Bailii},
  url = {http://www.bailii.org/ew/cases/EWCA/Civ/2009/889.html},
  urlyear = {23 April 2024}}
```

RX

📖 *Braceursself Ltd v NHS England*, 2013. England and Wales Court of Appeal (Civil Division), 39. *Construction law reports*, 212, pp.95–135.

⚙️ *Braceursself Ltd v NHS England*, 2013. England and Wales Court of Appeal (Civil Division), 39. *Construction law reports*, 212, pp.95–135.

```
@jurisdiction{braceursself-v-nhs.england2013,
  title = {{Braceursself} {Ltd} v-{NHS} {England}},
  year = {2013},
  institution = {England and Wales Court of Appeal (Civil Division)},
  number = {39},
  journal = {Construction law reports},
  volume = {212},
  pages = {95-135}}
```

RX

It is also possible to achieve a more traditional or **neutral citation** format, similar to Harvard (Bath)'s pre-2024 format:

📖 *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}}
```

RX

❗ Generally speaking, in a neutral citation 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.

💡 By default, 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},
```


❶ 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 = {2009},
note = {[2009] EWCA Civ 889}
```

References

- Adams, G., 2009. *Test citation one*. London: Imperial College Press.
- 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.
- Andrews, J. and Dury, A., 1773. *Map of Wiltshire*, 1 inch to 2 miles. Devizes: Wiltshire Record Society.
- The Archers, 2006. Radio. BBC Radio 4, 23 August. 19.02 hrs.
- Aristotle, 2007. *Nicomachean ethics* (W.D. Ross, Trans.). South Dakota: NuVisions.
- ASTM, 2019. *ASTM D1655 - 19 Standard specification for aviation turbine fuels*. West Conshohocken, Pa.: ASTM.
- Beethoven, L. van, 1950. *Symphony no.1 in C, Op.21*. Harmondsworth: Penguin.
- Blockley, D., 2021. *Engineering: a very short introduction* [Online]. Oxford: Oxford University Press. Available from: <https://www.amazon.co.uk/kindle> [Accessed 30 June 2021].
- Braceurself Ltd v NHS England*, 2013. England and Wales Court of Appeal (Civil Division), 39. *Construction law reports*, 212, pp.95–135.
- BRE, 2007. *Designing quality buildings: a BRE guide*. (Report 497). Bracknell: BRE.
- Bristol Region Building Record, 1965. *Green Park House (since demolished), viewed from southwest* [Photograph]. BRBR, D/877/1. Archives & Research Collections, University of Bath Library.
- British National Formulary, 2019. *Aspirin* [Online]. London: Pharmaceutical Press. Available from: https://www.medicinescomplete.com/#/content/bnf/_456850132 [Accessed 26 November 2019].
- British National Formulary, 2020a. 79th ed. *Aspirin*. London: Pharmaceutical Press, pp.280–281.
- British National Formulary, 2020b. 79th ed. London: Pharmaceutical Press.
- BSI, 1990. *BS 5605:1990 Recommendations for citing and referencing published material*. London: 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.
- Chakrabarti, V., 2016. *How architecture and city planning can combat social inequality* [Online]. Available from: <https://www.curbed.com/2016/5/5/11593058/vishaan-chakrabarti-pau-curbed-appeal-podcast> [Accessed 28 March 2019].
- ChatGPT, 2023. *Why is citing and referencing your sources important?* [Online]. San Francisco, Calif.: OpenAI. Available from: <https://chat.openai.com/share/782cb099-a0dc-45b1-8da4-0e99713f2d45> [Accessed 4 September 2023].
- Clark, T., 5 July 2004. A European UK Libraries Plus? *Lis-link* [Online]. Available from: lis-link@jiscmail.ac.uk [Accessed 30 July 2004].
- Cogley, M., 2020. Corporate confidence slumps to record low. *The Telegraph* [Online], 4 May. Available from: <https://www.telegraph.co.uk/business/2020/05/03/corporate-confidence-slumps-all-time-low/> [Accessed 5 May 2020].
- Council Regulation (EC) 1984/2003 of 8 April 2003 introducing a system for the statistical monitoring of trade in bluefin tuna, swordfish and big eye tuna within the Community [2003] *OJ* L295.
- 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.
- Deneulin, S. and Dinerstein, A.C., 2010. *Hope movements: Social movements in the pursuit of human development*. (Bath papers in international development and wellbeing, no. 8). Bath: University of Bath.
- Devlin, S.M., Martin, A. and Ostrovnaya, I., 2021. Identifying prognostic pairwise relationships among bacterial species in microbiome studies. *PLOS computational biology* [Online], 17(11). Available from: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009501> [Accessed 9 December 2021].
- The elephant man* [Online], 1980. Film. Directed by David Lynch. USA: Brookfilms. Available from: BBC iPlayer [Accessed 4 May 2021].
- 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.
- Gaiman, N., 2021. *Facebook post* [Online], 21 April. Available from: <https://www.facebook.com/neilgaiman/posts/305187897642814> [Accessed 13 May 2021].
- Ganju, V., 2021. *A study of EnGeneIC Dream Vectors (EDV's) packaged with the chemotherapy, E-EDV-D682 given simultaneously as non-targeted EDVs carrying an immune enhancer called EDV-GC, in participants with advanced pancreatic and other cancers whose disease has progressed after one or two treatment regimes, or where other standard therapies are not appropriate*. Sydney: University of Sydney. *Australian and New Zealand Clinical Trials Registry* [Online]. Available from: <http://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=365258> [Accessed 29 April 2021].
- Google, 2020. *Harbourside, Bristol* [Online], Google Maps. Available from: <https://www.google.co.uk/maps/place/Harbourside,+Bristol/> [Accessed 30 April 2020].
- Great Britain. Ministry of Defence, 2004. *Delivering security in a changing world: defence white paper*. (Cm. 6041). London: TSO.
- Great Britain. Parliament. House of Commons, 1988. *Local government finance bill*. (Bills | 1987/88, 66). London: HMSO.
- Great Britain. Parliament. House of Commons, 2004. *National Savings investment deposits: account 2002–2003*. (HC 2003/04, 30). London: National Audit Office.

- Great Britain. Parliament. House of Commons, 2024. Rwanda Plan Cost and Asylum System. *Hansard* [Online], 744, 9 January 2024. Available from: <https://hansard.parliament.uk/Commons/2024-01-09/debates/57105D77-72DE-4246-9F6E-ED87FCB9A088/RwandaPlanCostAndAsylumSystem> [Accessed 12 December 2024].
- Great Britain. Parliament. House of Lords, 1987. *Social fund (maternity and funeral expenses) bill*. (HL 1986/87, (66)). London: HMSO.
- Hadley, S., 2015. *Biomechanics: introductory reading, BM289: sport biomechanics*. University of Bath. Unpublished.
- Harris, G., 2013. *Focus group recommendations: internal task group report*. Unpublished.
- 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.hbcpnetbase.com> [Accessed 16 June 2016].
- 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/contents> [Accessed 22 April 2021].
- Hurry up and wait, 2021. *Inside no.9, Episode 6:4*. TV. BBC2, 31 May. 21.30 hrs.
- 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].
- Jones, J.W., n.d. a. *Dyfyniad prawf*. Caernarfon: Bwrdd yr Iaith Gymraeg.
- Jones, J.W., n.d. b. *Test citation*. Cardiff: Welsh Language Board.
- Library at University of Bath [@bathunilibary], 2021. *Twitter post* [Online], 10 May. Available from: <https://twitter.com/BathUniLibrary/status/1391774402618998795?s=20> [Accessed 12 May 2021].
- Liontou, C., Kontopodis, E., Oikonomidis, N., Maniotis, C., Tassopoulos, A., Tsiafoutis, I., Lazaris, E. and Koutouzis, M., 2019. Distal radial access: a review article. *Cardiovascular revascularization medicine* [Online], in press. Available from: <https://www.sciencedirect.com/science/article/pii/S1553838919303367> [Accessed 19 June 2019].
- Macbeth*, 1948. Film. Directed by Orson Welles. USA: Republic Pictures.
- Manco, J., 2013. *Vernacular architecture* [Online]. Available from: <https://www.buildinghistory.org/style/vernacular.shtml> [Accessed 20 February 2020].
- Moran, C., 2016. *Save our libraries* [Online]. Available from: <https://youtu.be/gKTfCz4JtVE> [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. *British medical journal*, 341(7765), pp.157–208.
- Open University, 1972. *Electricity and magnetism*. Bletchley: Open University Press.

- Ordnance Survey, 2020. *Street view map of University of Bath* [Online], 1:5000, OS VectorMap® Local. Available from: <https://digimap.edina.ac.uk/roam/map/os> [Accessed 30 April 2020].
- Pamporov, A., 2006. *Romskoto vsekidnevnie v Balgariya* [Roma everyday life in Bulgaria]. Veliko Tarnovo: Faber.
- 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].
- The secret, 2020. *Tiger king: Murder, mayhem and madness, Episode 3* [Online]. TV. Available from: Netflix [Accessed 4 May 2021].
- Seldon v Clarkson Wright & Jakes*, 2009. England and Wales Court of Appeal (Civil Division), 889. *Bailii* [Online]. Available from: <http://www.bailii.org/ew/cases/EWCA/Civ/2009/889.html> [Accessed 23 April 2024].
- 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].
- Smith, R., Jones, J.W., Stacey, D. and Lee, J., 2020. *Test nine*. Bath: Bath Publishing.
- Smith, R., Taylor, C., Rogers, T. and Wang, D., 2020. *Test eight*. Bath: Bath Publishing.
- Solomon, M.R., Askegaard, S., Hogg, M. and Bamossy, G.J., 2019. *Consumer behaviour: a European perspective*. 7th ed. Harlow: Pearson.
- 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.
- Steward, S., Connelly, D. and Robinson, J., 2020. Everything you should know about the coronavirus outbreak. *The pharmaceutical journal* [Online]. Available from: <https://www.pharmaceutical-journal.com/news-and-analysis/features/everything-you-should-know-about-the-coronavirus-outbreak/20207629.article> [Accessed 30 April 2020].
- 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.
- Thomas, D., 2015. Word count and referencing style. *Frequently asked questions discussion board: PHYS 2011: housing studies*. University of Bath. Unpublished.

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*. (PGI-93/WS/22). Paris: UNESCO.

University of Bath Library [@unibathlib], 2021. *Instagram post* [Online], 15 March. Available from: https://www.instagram.com/p/CMb5GHjFv9A/?utm_source=ig_web_copy_link [Accessed 7 May 2021].

Wang, J., 2018. *Test citation four*. New York: Random House.

Wang, J., Jones, J.W., Stacey, D. and Rogers, T., 2002. *Test six*. Bath: Really Decent Books.

Wang, K., 2020. *Test citation five*. New York: Random House.

Wang, K., Bloggs, C., Reilly, T. and Luscott, M., 2002. *Test seven*. Bath: Really Decent Books.

Wetzstein, G., Ozcan, A., Gigan, S., Fan, S., Englund, D., Soljacic, M., Denz, C., Miller, D.A.B. and Psaltis, D., 2020. Inference in artificial intelligence with deep optics and photonics. *Nature*, 588(7836), pp.39–47.

WHO (World Health Organization), 1986. *Nutrition and development in East Africa*. Geneva: WHO.

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].

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

World Health Organization, 2018. *The top 10 causes of death* [Online]. Geneva: World Health Organization. Available from: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death> [Accessed 29 June 2020].

6 Licence

Copyright 2016-2026 University of Bath.

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](#).

7 Implementation

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

Everything in this implementation is building towards section 7.9. That is where the `.bib` file is read, the entries are sorted, and the citation labels are generated. Once that is done, the code writes the `.bbl` bibliography file.

It starts by opening the `thebibliography` environment and providing various \LaTeX commands. It then uses the entry drivers in section 7.8 to write a `\bibitem` block for each entry. This block starts with natbib-flavoured citation data, followed by the formatted text to appear in the reference list. Lastly the environment is closed.

All the entry drivers start by calling the `output.bibitem` function, which is defined in section 7.7. This writes the `\bibitem` command, incorporating the generated label and adding some extra information. They then process the data using the name formats defined in section 7.5, the field formats defined in section 7.6 and the punctuation tracking and output functions defined in section 7.2.

7.1 Data model

We begin by listing the entry fields that we will recognize. These are the standard Bib \TeX ones plus `archive`, `doi`, `eid`, `eprint`, `entrysubtype`, `eventyear`, `keywords`, `language`, `library`, `options`, `pubstate`, `shortauthor`, `shorteditor`, `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   nameaddon
39   note
40   number
41   options
42   organization
43   pages
44   publisher
45   pubstate
46   school
47   series
48   shortauthor
49   shorteditor
50   sortyear
51   title
52   titleaddon
53   translator
54   type
55   url
56   urldate
57   urlyear
58   userb
```

```

59     venue
60     version
61     volume
62     year
63 }

```

The second and third arguments to `ENTRY` are integer and string fields (respectively) for which the style will generate its own values.

```

64 { cite.initials disamb.names }
65 { label label.source extra.label sort.label short.list }

```

7.2 Utilities for writing reference entries to the file

This section contains generic functions for writing out the formatted reference.

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

```

66 INTEGERS {
67     output.state before.all mid.clause mid.sentence after.sentence after.block
68     online.shown
69 }
70 FUNCTION {init.state.consts}
71 { #0 'before.all :=
72   #1 'mid.clause :=
73   #2 'mid.sentence :=
74   #3 'after.sentence :=
75   #4 'after.block :=
76 }

```

We also define some variables to store strings in.

```

77 STRINGS { s t field abbrev }

```

The `output.nonnull` function takes the top two tokens (strings) from the stack. It appends the appropriate delimiter to the earlier token and writes it to the `.bbl` file. It then returns the later (topmost) token to the top of the stack ready for the next occurrence of the function. These are the delimiters used in each state:

- `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.

Afterwards, the output state is set to `mid.sentence`.

```

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

```



```

94         if$
95     }
96     if$
97     mid.sentence 'output.state :=
98 }
99 if$
100 s
101 }

```

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’.

```

102 FUNCTION {end.clause}
103 { output.state mid.clause =
104   { mid.sentence 'output.state := }
105   'skip$
106   if$
107 }

```

We provide two wrappers around `output.nonnull` to avoid passing it empty strings: `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.

```

108 FUNCTION {output}
109 { duplicate$ empty$
110   { pop$ end.clause }
111   'output.nonnull
112   if$
113 }
114 FUNCTION {output.check}
115 { 't :=
116   duplicate$ empty$
117   { pop$ "empty " t * " in " * cite$ * warning$
118     end.clause
119   }
120   'output.nonnull
121   if$
122 }

```

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`.

```

123 FUNCTION {new.block}
124 { output.state before.all =
125   'skip$
126   { after.block 'output.state := }
127   if$
128 }
129 FUNCTION {new.sentence}
130 { output.state after.block =
131   'skip$
132   { output.state before.all =
133     'skip$
134     { after.sentence 'output.state := }
135     if$
136   }
137   if$
138 }
139 FUNCTION {continue.clause}
140 { output.state after.sentence =
141   'skip$
142   { output.state after.block =
143     'skip$
144     { output.state before.all =
145       'skip$
146       { mid.clause 'output.state := }
147       if$
148     }
149     if$
150   }
151   if$

```


152 }

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

```
153 FUNCTION {date.block}
154 { new.block
155 }
```

The `bibinfo` functions consume two tokens – the value and name of a field respectively – and return a single string. If the value is non-null, the tokens are concatenated in the form `\bibinfo{field name}{field value}`, otherwise an empty string is returned.

The `check` version does this silently, while the `warn` version issues a warning in the log file if the value is missing or empty.

```
156 FUNCTION {bibinfo.check}
157 { swap$
158   duplicate$ missing$
159   { pop$ pop$
160     ""
161   }
162   { duplicate$ empty$
163     { swap$ pop$
164       }
165     { swap$
166       "\bibinfo{" swap$ * "}-{ " * swap$ * "}" *
167     }
168     if$
169   }
170   if$
171 }
172 FUNCTION {bibinfo.warn}
173 { swap$
174   duplicate$ missing$
175   { swap$ "missing " swap$ * " in " * cite$ * warning$ pop$
176     ""
177   }
178   { duplicate$ empty$
179     { swap$ "empty " swap$ * " in " * cite$ * warning$
180       }
181     { swap$
182       "\bibinfo{" swap$ * "}-{ " * swap$ * "}" *
183     }
184     if$
185   }
186   if$
187 }
```

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.

```
188 FUNCTION {fin.entry}
189 { add.period$
190   write$
191   newline$
192 }
```

7.3 Convenience functions

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

```
194 FUNCTION {not}
195 { { #0 }
196   { #1 }
197   if$
198 }
199 FUNCTION {and}
200 { 'skip$
201   { pop$ #0 }
202   if$
```

```

203 }
204 FUNCTION {or}
205 { { pop$ #1 }
206   'skip$
207   if$
208 }

```

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

```

209 FUNCTION {new.block.checkb}
210 { empty$
211   swap$ empty$
212   and
213   'skip$
214   'new.block
215   if$
216 }

```

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

```

217 FUNCTION {field.or.null}
218 { duplicate$ empty$
219   { pop$ "" }
220   'skip$
221   if$
222 }

```

The `emphasize` function applies emphasis to the topmost token.

```

223 FUNCTION {emphasize}
224 { duplicate$ empty$
225   { pop$ "" }
226   { "\emph{" swap$ * "}" * }
227   if$
228 }

```

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.

```

229 FUNCTION {tie.or.space.prefix}
230 { duplicate$ text.length$ #3 <
231   { "~" }
232   { " " }
233   if$
234   swap$
235 }

```

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

```

236 FUNCTION {capitalize}
237 { "u" change.case$ "t" change.case$ }

```

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

```

238 FUNCTION {space.word}
239 { " " swap$ * " " * }

```

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

```

240 FUNCTION {either.or.check}
241 { empty$
242   'pop$
243   { "can't use both " swap$ * " fields in " * cite$ * warning$ }
244   if$
245 }

```

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

```

246 FUNCTION {is.online}
247 { online.shown
248   { #0 }
249   { url empty$
250     doi empty$ and
251     urldate empty$ and
252     urlyear empty$ and
253     { type$ "online" =
254       { #1 }
255       { #0 }
256       if$
257     }
258     { #1 }
259     if$
260   }
261   if$
262 }

```

7.4 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.

```

263 FUNCTION {bbl.and}
264 { "and" }
265
266 FUNCTION {bbl.byeditor}
267 { "Ed. by" }
268
269 FUNCTION {bbl.etal}
270 { "et~al." }
271
272 FUNCTION {bbl.editors}
273 { "eds" }
274
275 FUNCTION {bbl.editor}
276 { "ed." }
277
278 FUNCTION {bbl.edby}
279 { "edited by" }
280
281 FUNCTION {bbl.translator}
282 { "Trans." }
283
284 FUNCTION {bbl.edition}
285 { "ed." }
286
287 FUNCTION {bbl.volume}
288 { "vol." }
289
290 FUNCTION {bbl.of}
291 { "of" }
292
293 FUNCTION {bbl.number}
294 { "no." }
295
296 FUNCTION {bbl.nr}
297 { "no." }
298
299 FUNCTION {bbl.in}
300 { "in" }
301
302 FUNCTION {bbl.pages}
303 { "pp." }
304
305 FUNCTION {bbl.page}
306 { "p." }
307
308 FUNCTION {bbl.chapter}
309 { "chap." }
310

```

```

311 FUNCTION {bbl.techrep}
312 { "Tech. Rep." }
313
314 FUNCTION {bbl.mthesis}
315 { "Master's thesis" }
316
317 FUNCTION {bbl.phdthesis}
318 { "Ph.D. thesis" }
319
320 FUNCTION {bbl.first}
321 { "1st" }
322
323 FUNCTION {bbl.second}
324 { "2nd" }
325
326 FUNCTION {bbl.third}
327 { "3rd" }
328
329 FUNCTION {bbl.fourth}
330 { "4th" }
331
332 FUNCTION {bbl.fifth}
333 { "5th" }
334
335 FUNCTION {bbl.st}
336 { "st" }
337
338 FUNCTION {bbl.nd}
339 { "nd" }
340
341 FUNCTION {bbl.rd}
342 { "rd" }
343
344 FUNCTION {bbl.th}
345 { "th" }
346
347 FUNCTION {bbl.commission}
348 { "Commission Decision" }
349
350 FUNCTION {bbl.online}
351 { "[Online]" }
352
353 FUNCTION {bbl.nodate}
354 { "n.d." }
355
356 FUNCTION {bbl.inpreparation}
357 { "preprint" }
358
359 FUNCTION {bbl.submitted}
360 { "preprint" }
361
362 FUNCTION {bbl.inpress}
363 { "in press" }
364
365 FUNCTION {bbl.unpublished}
366 { "Unpublished" }

```

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

```

368 MACRO {jan} {"January"}
369
370 MACRO {feb} {"February"}
371
372 MACRO {mar} {"March"}
373
374 MACRO {apr} {"April"}
375
376 MACRO {may} {"May"}
377
378 MACRO {jun} {"June"}
379
380 MACRO {jul} {"July"}

```

```

381
382 MACRO {aug} {"August"}
383
384 MACRO {sep} {"September"}
385
386 MACRO {oct} {"October"}
387
388 MACRO {nov} {"November"}
389
390 MACRO {dec} {"December"}

```

This function takes a cardinal number and replaces it with the corresponding English ordinal.

```

392 FUNCTION {eng.ord}
393 { duplicate$ "1" swap$ *
394   #-2 #1 substring$ "1" =
395   { bbl.th * }
396   { duplicate$ #-1 #1 substring$
397     duplicate$ "1" =
398     { pop$ bbl.st * }
399     { duplicate$ "2" =
400       { pop$ bbl.nd * }
401       { "3" =
402         { bbl.rd * }
403         { bbl.th * }
404         if$
405       }
406       if$
407     }
408     if$
409   }
410   if$
411 }

```

word.in provides the ‘In:’ before book titles.

```

412 FUNCTION {word.in}
413 { bbl.in capitalize
414   ":" *
415   " " * }

```

7.5 Name formats

These variables will help with iterating through the names.

```

417 INTEGERS { nameptr namesleft numnames }

```

7.5.1 Names in verbose citations

These name formats support natbib’s `\citet*` and `\citep*` commands, which suppress the normal ‘et al.’ truncation so all author surnames are shown.

`format.full.names` consumes one token (a name field) and returns a list of surnames. The ‘full’ bit refers to the list, rather than the individual names included in it. The first name may appear as it does in the reference list (that is, followed by the initial) to distinguish it from a different lead author with the same surname.

```

418 FUNCTION {format.full.names}
419 { 's :=
420   "" 't :=
421   #1 'nameptr :=
422   s num.names$ 'numnames :=
423   numnames 'namesleft :=
424   { namesleft #0 > }
425   { s nameptr
426     duplicate$ #1 =
427     cite.initials
428     and
429     { "{11}{, jj}{, f{.}.}{~vv}" }
430     { "{11}" }

```

```

431     if$
432     format.name$
433     't :=
434     nameptr #1 >
435     { namesleft #1 >
436       { ", " * t * }
437       { s nameptr "{11}" format.name$ duplicate$ "others" =
438         { 't := }
439         { pop$ }
440         if$
441         t "others" =
442         { " " * bbl.etal *
443         }
444         { bbl.and
445           space.word * t *
446         }
447         if$
448       }
449       if$
450     }
451     't
452     if$
453     nameptr #1 + 'nameptr :=
454     namesleft #1 - 'namesleft :=
455   }
456   while$
457 }

```

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.

```

458 FUNCTION {author.editor.key.full}
459 { shortauthor empty$
460   { author empty$
461     { shorteditor empty$
462       { editor empty$
463         { key empty$
464           { cite$ #1 #3 substring$ }
465           'key
466           if$
467         }
468         { editor format.full.names }
469         if$
470       }
471       { shorteditor format.full.names }
472       if$
473     }
474     { author format.full.names }
475     if$
476   }
477   { shortauthor format.full.names }
478   if$
479 }

```

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

```

480 FUNCTION {editor.key.full}
481 { shorteditor empty$
482   { editor empty$
483     { key empty$
484       { cite$ #1 #3 substring$ }
485       'key
486       if$
487     }
488     { editor format.full.names }
489     if$
490   }
491   { shorteditor format.full.names }
492   if$

```

```
493 }
```

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

```
494 FUNCTION {author.key.full}
495 { shortauthor empty$
496   { author empty$
497     { key empty$
498       { cite$ #1 #3 substring$ }
499       'key
500       if$
501     }
502     { author format.full.names }
503     if$
504   }
505   { shortauthor format.full.names }
506   if$
507 }
```

make.full.names chooses between the above.

```
508 FUNCTION {make.full.names}
509 { type$ "book" =
510   type$ "inbook" =
511   or
512   'author.editor.key.full
513   { type$ "proceedings" =
514     'editor.key.full
515     'author.key.full
516     if$
517   }
518   if$
519 }
```

7.5.2 Names in normal citations

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 later names are replaced with ‘et al.’ The number of names remaining equals the entry’s disamb.names value (default 0) plus 1; this gets set by the functions in section 7.9.2.

```
520 FUNCTION {format.lab.names}
521 { 's :=
522   "" 't :=
523   #1 'nameptr :=
524   s num.names$ 'numnames :=
525   numnames 'namesleft :=
526   { namesleft #0 > }
527   { s nameptr
528     duplicate$ #1 =
529     cite.initials
530     and
531     { "{ll}{, jj}{, f{.}.}{~vv}" }
532     { "{ll}" }
533     if$
534     format.name$
535     't :=
536     nameptr #1 >
537     { nameptr #2 disamb.names + =
538       numnames #3 > and
539       { "others" 't :=
540         #1 'namesleft := }
541       'skip$
542     if$
543     namesleft #1 >
544     { ", " * t * }
545     { s nameptr "{ll}" format.name$ duplicate$ "others" =
546       { 't := }
547       { pop$ }
548     if$
```

```

549         t "others" =
550         { " " * bbl.etal *
551         }
552         { bbl.and
553         space.word * t *
554         }
555         if$
556     }
557     if$
558 }
559 't
560 if$
561 nameptr #1 + 'nameptr :=
562 namesleft #1 - 'namesleft :=
563 }
564 while$
565 }

```

title.label gives the title with the correct font shape, for use as a fallback label. This needs to be kept in sync with the logic used in the various drivers and title formatting functions.

```

566 FUNCTION {title.label}
567 { title
568   duplicate$ 'label.source :=
569   "t" change.case$
570   type$ duplicate$
571   "book" = swap$
572   "collection" =
573   or
574   'emphasize
575   {
576     type$ duplicate$
577     "techreport" = swap$
578     "report" =
579     or
580     'emphasize
581     {
582       type$ duplicate$ duplicate$ duplicate$
583       duplicate$ duplicate$ duplicate$ duplicate$
584       "manual" = swap$
585       "dataset" =
586       or swap$
587       "electronic" =
588       or swap$
589       "online" =
590       or swap$
591       "patent" =
592       or swap$
593       "software" =
594       or swap$
595       "standard" =
596       or swap$
597       "www" =
598       or
599       'emphasize
600       {
601         type$ duplicate$ duplicate$ duplicate$ duplicate$
602         "booklet" = swap$
603         "audio" =
604         or swap$
605         "movie" =
606         or swap$
607         "music" =
608         or swap$
609         "video" =
610         or
611         { type duplicate$ empty$
612           { pop$ emphasize }
613           { duplicate$ "TV" = swap$ "Radio" = or
614             'skip$
615             'emphasize
616             if$

```



```

617     }
618     if$
619   }
620   {
621     type$ "image" =
622       'emphasize
623     {
624       type$ "jurisdiction" =
625         'emphasize
626       {
627         type$ "legislation" =
628           { journal empty$
629             'emphasize
630             'skip$
631             if$
632           }
633           {
634             type$ "proceedings" =
635               'emphasize
636             {
637               type$ "unpublished" =
638                 booktitle empty$
639                 and
640                 'emphasize
641                 {}
642                 if$
643               } if$
644             } if$
645           } if$
646         } if$
647       } if$
648     } if$
649   } if$
650 } if$
651 }

```

The following three functions enable the label to be chosen from several fields. In each case the raw data used is saved in the entry's `label.source` field, so that we can tell by comparison whether it is a name list or not later on.

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

```

652 FUNCTION {author.editor.key.label}
653 { shortauthor empty$
654   { author empty$
655     { shorteditor empty$
656       { editor empty$
657         { key empty$
658           { title empty$
659             { cite$
660               duplicate$ 'label.source :=
661               #1 #3 substring$
662             }
663             'title.label
664             if$
665           }
666           { key
667             duplicate$ 'label.source :=
668           }
669           if$
670         }
671         { editor
672           duplicate$ 'label.source :=
673           format.lab.names
674         }
675       } if$
676     }
677     { shorteditor
678       duplicate$ 'label.source :=
679       format.lab.names
680     }

```

```

681         if$
682     }
683     { author
684         duplicate$ 'label.source :=
685         format.lab.names
686     }
687     if$
688 }
689 { shortauthor
690     duplicate$ 'label.source :=
691     format.lab.names
692 }
693 if$
694 }

```

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

```

695 FUNCTION {editor.key.label}
696 { shorteditor empty$
697   { editor empty$
698     { key empty$
699       { title empty$
700         { cite$
701           duplicate$ 'label.source :=
702           #1 #3 substring$
703         }
704         'title.label
705         if$
706       }
707       { key
708         duplicate$ 'label.source :=
709       }
710       if$
711     }
712     { editor
713       duplicate$ 'label.source :=
714       format.lab.names
715     }
716     if$
717   }
718   { shorteditor
719     duplicate$ 'label.source :=
720     format.lab.names
721   }
722   if$
723 }

```

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

```

724 FUNCTION {author.key.label}
725 { shortauthor empty$
726   { author empty$
727     { key empty$
728       { type$ "inreference" =
729         booktitle empty$ not
730         and
731         { booktitle
732           duplicate$ 'label.source :=
733         }
734         { title empty$
735           { cite$
736             duplicate$ 'label.source :=
737             #1 #3 substring$
738           }
739           'title.label
740           if$
741         }
742         if$
743       }
744     { key

```

```

745         duplicate$ 'label.source :=
746     }
747     if$
748 }
749 { author
750     duplicate$ 'label.source :=
751     format.lab.names
752 }
753 if$
754 }
755 { shortauthor
756     duplicate$ 'label.source :=
757     format.lab.names
758 }
759 if$
760 }

```

calc.short.authors chooses between the above.

```

761 FUNCTION {calc.short.authors}
762 { type$ "book" =
763   type$ "inbook" =
764   or
765     'author.editor.key.label
766     { type$ "proceedings" =
767       'editor.key.label
768       'author.key.label
769       if$
770     }
771   if$
772   'short.list :=
773 }

```

7.5.3 Names at the head of the reference

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.

```

774 FUNCTION {format.names}
775 { 'field :=
776   duplicate$ empty$
777   'skip$
778   { 's :=
779     "" 't :=
780     #1 'nameptr :=
781     s num.names$ 'numnames :=
782     numnames 'namesleft :=
783     { namesleft #0 > }
784     { s nameptr
785       "{ll}{, jj}{, f{.}.}{~vv}"
786       format.name$
787       field bibinfo.check
788       't :=
789       nameptr #1 >
790       { namesleft #1 >
791         { ", " * t * }
792         { s nameptr "{ll}" format.name$ duplicate$ "others" =
793           { 't := }
794           { pop$ }
795           if$
796           t "others" =
797           { " " * bbl.etal *
798             }
799           { bbl.and
800             space.word * t *
801             }
802           if$
803         }
804         if$
805       }

```

```

806         't
807         if$
808             nameptr #1 + 'nameptr :=
809             namesleft #1 - 'namesleft :=
810         }
811     while$
812 }
813 if$
814 }

```

`format.names.short` consumes three tokens (field, short field, string field name) and operates much like `format.names`, except that the field (e.g. `author`) and its short counterpart (e.g. `shortauthor`) are iterated in parallel. If the values differ, the short version is printed plain and the regular version is printed after in parentheses. If the short field is empty or does not have the same number of names as the regular field, the regular field is processed with `format.names` instead.

```

815 FUNCTION {format.names.short}
816 { 'field :=
817   duplicate$ empty$
818   { pop$ field format.names }
819   { 'abbrev :=
820     duplicate$ empty$
821     'skip$
822     { 's :=
823       "" 't :=
824       #1 'nameptr :=
825       s num.names$ duplicate$ 'numnames :=
826       abbrev num.names$ =
827       { numnames 'namesleft :=
828         { namesleft #0 > }
829         { abbrev nameptr "{1l}{, jj}{, f{.}.}{~vv}" format.name$
830           't :=
831           s nameptr "{1l}{, jj}{, f{.}.}{~vv}" format.name$
832           duplicate$ t =
833           { field bibinfo.check 't := }
834           { " (" swap$ * ")" *
835             t field bibinfo.check swap$ *
836             't :=
837           }
838           if$
839           nameptr #1 >
840           { namesleft #1 >
841             { ", " * t * }
842             { s nameptr "{1l}" format.name$ duplicate$ "others" =
843               { 't := }
844               { pop$ }
845               if$
846               t "others" =
847               { " " * bbl.etal *
848                 }
849               { bbl.and
850                 space.word * t *
851               }
852               if$
853             }
854             if$
855           }
856           't
857           if$
858           nameptr #1 + 'nameptr :=
859           namesleft #1 - 'namesleft :=
860         }
861         while$
862       }
863       { s field format.names }
864     if$
865   }
866   if$
867 }
868 if$
869 }

```

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

```

870 FUNCTION {format.names.ed}
871 { 'field :=
872   duplicate$ empty$
873   'skip$
874   { 's :=
875     "" 't :=
876     #1 'nameptr :=
877     s num.names$ 'numnames :=
878     numnames 'namesleft :=
879     { namesleft #0 > }
880     { s nameptr
881       "{f{.}.~}{vv~}{ll}{ jj}"
882       format.name$
883       field bibinfo.check
884       't :=
885       nameptr #1 >
886       { namesleft #1 >
887         { ", " * t * }
888         { s nameptr "{ll}" format.name$ duplicate$ "others" =
889           { 't := }
890           { pop$ }
891           if$
892           t "others" =
893           { " " * bbl.etal *
894             }
895           { bbl.and
896             space.word * t *
897             }
898           if$
899           }
900           if$
901           }
902           't
903           if$
904           nameptr #1 + 'nameptr :=
905           namesleft #1 - 'namesleft :=
906           }
907           while$
908           }
909           if$
910 }

```

`format.authors` returns author names.

```

911 FUNCTION {format.authors}
912 { author shortauthor "author" format.names.short
913 }

```

`format.byeditors` returns editor names, and prepended by ‘Ed. by’.

```

914 FUNCTION {format.byeditors}
915 { editor "editor" format.names.ed duplicate$ empty$ 'skip$
916   { bbl.byeditor
917     " " *
918     swap$ *
919   }
920   if$
921 }

```

`format.editors` returns editor names, and appends ‘ed.’ or ‘eds’ as appropriate.

```

922 FUNCTION {get.bbl.editor}
923 { editor num.names$ #1 > 'bbl.editors 'bbl.editor if$ }
924 FUNCTION {format.editors}
925 { editor shorteditor "editor" format.names.short duplicate$ empty$ 'skip$
926   { ", " *
927     " " *
928     get.bbl.editor

```

```

929     *
930   }
931   if$
932 }

```

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

```

933 FUNCTION {format.translators}
934 { translator "translator" format.names.ed duplicate$ empty$ 'skip$
935   { "," *
936     " " *
937     bbl.translator *
938     "(" swap$ *
939     ")" *
940   }
941   if$
942 }

```

7.6 Field formats

Each of the `format.*` functions below (and above) add a single string (possibly empty) to the stack, based on the value of one or more fields.

7.6.1 Head block

`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 an extra defence against an entry beginning with the date.

```

943 FUNCTION {format.key}
944 { empty$
945   { key field.or.null }
946   { "" }
947   if$
948   nameaddon duplicate$ empty$
949   'pop$
950   { swap$ duplicate$ empty$
951     'skip$
952     { " " * }
953     if$
954     swap$ "[" swap$ * * "]" *
955     continue.clause
956   }
957   if$
958 }

```

7.6.2 Title block

`select.language` checks if the string at the top of the stack is empty, and if not, wraps `{\selectlanguage{language}...}` around it.

```

959 FUNCTION {select.language}
960 { duplicate$ empty$
961   'pop$
962   { language empty$
963     'skip$
964     { "{\selectlanguage{" language * "}" * swap$ * "}" * }
965     if$
966   }
967   if$
968 }

```

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

```

969 FUNCTION {format.title}
970 { title
971   duplicate$ empty$ 'skip$

```

```

972     { "t" change.case$ }
973   if$
974   "title" bibinfo.check
975   duplicate$ empty$
976   'skip$
977   { select.language }
978   if$
979 }

```

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

```

980 FUNCTION {format.btitle}
981 { title
982   duplicate$ empty$ 'skip$
983   { "t" change.case$ }
984   if$
985   "title" bibinfo.check
986   duplicate$ empty$
987   'skip$
988   { emphasize select.language }
989   if$
990 }

```

`format.avtitle` returns the title and language fields, where the shape of the title (upright or italic) depends on the type.

```

991 FUNCTION {format.avtitle}
992 { title
993   duplicate$ empty$ 'skip$
994   { "t" change.case$ }
995   if$
996   "title" bibinfo.check
997   duplicate$ empty$
998   'skip$
999   { type duplicate$ empty$
1000     { pop$ emphasize }
1001     { duplicate$ "TV" = swap$ "Radio" = or
1002       'skip$
1003       'emphasize
1004       if$
1005     }
1006     if$
1007     select.language
1008   }
1009   if$
1010 }

```

`format.titleaddon` returns the titleaddon field. This is the `bath.bst` version:

```

1011 FUNCTION {format.titleaddon}
1012 { titleaddon
1013   duplicate$ empty$ 'skip$
1014   { "titleaddon" bibinfo.check }
1015   if$
1016 }

```

This is the `bathx.bst` version:

```

1011 FUNCTION {format.titleaddon}
1012 { titleaddon
1013   duplicate$ empty$ 'skip$
1014   { "[" swap$ "titleaddon" bibinfo.check * "]" * }
1015   if$
1016 }

```

`format.version` returns the version field.

```

1017 FUNCTION {format.version}
1018 { version
1019   duplicate$ empty$ 'skip$

```

```

1020 { "(v." swap$ *
1021 "version" bibinfo.check
1022 ")" *
1023 }
1024 if$
1025 }

```

`print.labeltitle.addenda` is a convenience for writing out the elements that normally follow the title when it appears at the head of the reference.

```

1026 FUNCTION {print.labeltitle.addenda}
1027 { continue.clause

```

In `bath.bst` it prints the version and then `titleaddon` field:

```

1028 online.shown
1029 'skip$
1030 { format.version output
1031   continue.clause
1032   format.titleaddon output
1033   #1 'online.shown :=
1034 }
1035 if$

```

In `bathx.bst` it prints the `titleaddon` field, then the version, then either the `entrysubtype` or the `bbl.online` string as appropriate:

```

1028 format.titleaddon output
1029 continue.clause
1030 type$ duplicate$ "legislation" = not
1031 swap$ "jurisdiction" = not
1032 and
1033 { format.version output
1034   is.online
1035   { entrysubtype duplicate$ empty$
1036     { pop$
1037       journal empty$
1038       booktitle empty$
1039       type$ "inreference" =
1040       author empty$
1041       and
1042       or
1043       and
1044       library empty$
1045       type$ duplicate$ "image" =
1046       swap$ duplicate$ "audio" =
1047       swap$ duplicate$ "video" =
1048       swap$ duplicate$ "music" =
1049       swap$ "movie" =
1050       or
1051       or
1052       or
1053       or
1054       or
1055       and
1056       series empty$
1057       type field.or.null
1058       duplicate$ "TV" =
1059       swap$ "Radio" =
1060       or not
1061       or
1062       and
1063       { continue.clause
1064         bbl.online output
1065         #1 'online.shown :=
1066       }
1067       'skip$
1068     if$
1069   }
1070   { "entrysubtype" bibinfo.check
1071     "[" swap$ * "]" *

```



```

1072         continue.clause output
1073     }
1074     if$
1075 }
1076 'skip$
1077 if$
1078 }
1079 'skip$
1080 if$

```

`print.title.addenda` is a convenience for writing out the elements that normally follow the title in its regular position. It is similar to `print.labeltitle.addenda` but has somewhat different logic.

```

1081 }
1082 FUNCTION {print.title.addenda}
1083 { continue.clause

```

In `bath.bst` it prints the version and then `titleaddon` field:

```

1039 online.shown
1040 'skip$
1041 { format.version output
1042   continue.clause
1043   format.titleaddon output
1044   #1 'online.shown :=
1045 }
1046 if$

```

In `bathx.bst` it prints the `titleaddon` field, then the version, then either the `entrysubtype` or the `bb1.online` string as appropriate:

```

1084 type$ duplicate$ "legislation" = not
1085 swap$ "jurisdiction" = not
1086 and
1087 { format.titleaddon output
1088   continue.clause
1089 }
1090 'skip$
1091 if$
1092 format.version output
1093 is.online
1094 { entrysubtype duplicate$ empty$
1095   type$ duplicate$ "legislation" =
1096   swap$ duplicate$ "jurisdiction" =
1097   swap$ "legal" =
1098   or
1099   or
1100   or
1101   { pop$
1102     journal empty$
1103     booktitle empty$
1104     type$ "inreference" =
1105     author empty$
1106     and
1107     or
1108     and
1109     library empty$
1110     type$ duplicate$ "image" =
1111     swap$ duplicate$ "audio" =
1112     swap$ duplicate$ "video" =
1113     swap$ duplicate$ "music" =
1114     swap$ "movie" =
1115     or
1116     or
1117     or
1118     or
1119     or
1120     and
1121     series empty$
1122     type field.or.null

```

```

1123         duplicate$ "TV" =
1124         swap$ "Radio" =
1125         or not
1126     or
1127     and
1128     { continue.clause
1129         bbl.online output
1130         #1 'online.shown :=
1131     }
1132     'skip$
1133     if$
1134 }
1135 { "entrysubtype" bibinfo.check
1136     "[" swap$ * "]" *
1137     continue.clause output
1138 }
1139     if$
1140 }
1141 'skip$
1142 if$

```

We finish off with editor/translator information.

```

1143 continue.clause
1144 type$ "reference" =
1145     { format.byeditors output }
1146     'skip$
1147 if$
1148 format.translators output
1149 }

```

format.label.booktitle returns the booktitle field upright

```

1150 FUNCTION {format.label.booktitle}
1151 { booktitle
1152     duplicate$ empty$ 'skip$
1153     { "t" change.case$ }
1154     if$
1155     "booktitle" bibinfo.check
1156 }

```

format.booktitle returns the booktitle field in italics.

```

1157 FUNCTION {format.booktitle}
1158 { format.label.booktitle
1159     emphasize
1160 }

```

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

```

1161 FUNCTION {format.in.ed.booktitle}
1162 { format.booktitle
1163     duplicate$ empty$
1164     'skip$
1165     {

```

In bathx.bst, we conditionally include the bbl.online string:

```

1166 is.online
1167 { " " *
1168     bbl.online *
1169     #1 'online.shown :=
1170 }
1171 'skip$
1172 if$

```

```

1173 editor "editor" format.names.ed duplicate$ empty$ 'pop$
1174 { "," *
1175     " " *
1176     get.bbl.editor add.period$
1177     " " *

```

```

1178     * swap$
1179     *
1180     word.in swap$ *
1181   }
1182   if$
1183 }
1184 if$
1185 }

```

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

```

1186 FUNCTION {legal.journal.shape}
1187 { duplicate$ "OJ" =
1188   'emphasize
1189   'skip$
1190   if$
1191 }

```

7.6.3 Dates

In `bath.bst`, `nodate.check` doesn’t do anything:

```

1091 FUNCTION {nodate.check}
1092 {
1093   skip$
1094 }

```

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

```

1192 FUNCTION {nodate.check}
1193 {
1194   duplicate$ empty$
1195   sortyear empty$
1196   and
1197   { options field.or.null "nodate" =
1198     'skip$
1199     { pop$ bbl.nodate }
1200     if$
1201   }
1202   'skip$
1203   if$
1204 }

```

`format.date` returns the year and any disambiguation label.

```

1205 FUNCTION {format.date}
1206 { year "year" bibinfo.check nodate.check
1207   extra.label *
1208 }

```

`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.

```

1209 FUNCTION {format.date.brackets}
1210 { format.date
1211   "[" swap$ * "]" *
1212 }
1213 FUNCTION {format.date.parens}
1214 { format.date
1215   "(" swap$ * ")" *
1216 }
1217 FUNCTION {format.jur.date.essential}
1218 { options field.or.null "scottish-style" =

```

```

1219 keywords field.or.null "sc" =
1220 or
1221 { end.clause format.date }
1222 'format.date.brackets
1223 if$
1224 }
1225 FUNCTION {format.jur.date}
1226 { institution empty$
1227   number empty$
1228   or
1229   { journal field.or.null 's :=
1230     s "OJ" =
1231     s "ECR" =
1232     or
1233     'format.date.brackets
1234     { options field.or.null "year-essential=false" =
1235       'format.date.parens
1236       { volume empty$
1237         'format.jur.date.essential
1238         { options field.or.null "year-essential=true" =
1239           'format.jur.date.essential
1240           'format.date.parens
1241           if$
1242         }
1243       }
1244     }
1245   }
1246 }
1247 if$
1248 }
1249 { end.clause format.date }
1250 if$
1251 }

```

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

```

1252 FUNCTION {format.eu.date}
1253 { journal field.or.null 's :=
1254   s "OJ" =
1255   s "ECR" =
1256   or
1257   'format.date.brackets
1258   'format.date
1259   if$
1260 }

```

7.6.4 Series, types and numbering

`format.series` returns the case-altered series field.

```

1261 FUNCTION {format.series}
1262 { series
1263   duplicate$ empty$ 'skip$
1264   { "t" change.case$ }
1265   if$
1266   "series" bibinfo.check
1267 }

```

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

```

1268 FUNCTION {format.bvolume}
1269 { volume empty$
1270   { "" }
1271   { bbl.volume volume tie.or.space.prefix
1272     "volume" bibinfo.check * *
1273     format.series
1274     duplicate$ empty$ 'pop$
1275     { emphasize ", " * swap$ * }
1276     if$

```

```

1277     "volume and number" number either.or.check
1278   }
1279   if$
1280 }

```

`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.

```

1281 FUNCTION {format.number.series}
1282 { volume empty$
1283   { number empty$
1284     { series field.or.null }
1285     { series empty$
1286       { number "number" bibinfo.check }
1287       { format.series
1288         number tie.or.space.prefix "number" bibinfo.check * *
1289       }
1290     }
1291   }
1292   if$
1293 }
1294 { "" }
1295 if$
1296 }

```

`format.series.episode` returns the case-altered series and the number, separated by a space and both emphasised.

```

1297 FUNCTION {format.series.episode}
1298 { series
1299   duplicate$ empty$
1300   'skip$
1301   { "t" change.case$ }
1302   if$
1303   "series" bibinfo.check emphasize
1304   number "number" bibinfo.check emphasize
1305   duplicate$ empty$
1306   { * }
1307   { swap$
1308     duplicate$ empty$
1309     { * }
1310     { ", " * swap$ * }
1311   }
1312   if$
1313 }
1314 }

```

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

```

1315 FUNCTION {is.num}
1316 { chr.to.int$
1317   duplicate$ "0" chr.to.int$ < not
1318   swap$ "9" chr.to.int$ > not and
1319 }

```

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

```

1320 FUNCTION {extract.num}
1321 { duplicate$ 't :=
1322   "" 's :=
1323   { t empty$ not }
1324   { t #1 #1 substring$
1325     t #2 global.max$ substring$ 't :=
1326     duplicate$ is.num
1327     { s swap$ * 's := }
1328     { pop$ "" 't := }
1329   }
1330   if$
1331 }
1332 while$
1333 s empty$

```

```

1333     'skip$
1334     { pop$ s }
1335     if$
1336 }

```

`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.

```

1337 FUNCTION {convert.edition}
1338 { extract.num "1" change.case$ 's :=
1339   s "first" = s "1" = or
1340   { bbl.first 't := }
1341   { s "second" = s "2" = or
1342     { bbl.second 't := }
1343     { s "third" = s "3" = or
1344       { bbl.third 't := }
1345       { s "fourth" = s "4" = or
1346         { bbl.fourth 't := }
1347         { s "fifth" = s "5" = or
1348           { bbl.fifth 't := }
1349           { s #1 #1 substring$ is.num
1350             { s eng.ord 't := }
1351             { edition 't := }
1352             if$
1353           }
1354         if$
1355       }
1356     if$
1357   }
1358   if$
1359 }
1360 if$
1361 }
1362 if$
1363 t
1364 }

```

`format.edition` returns the converted edition field and ‘ed.’

```

1365 FUNCTION {format.edition}
1366 { edition duplicate$ empty$ 'skip$
1367   { convert.edition
1368     output.state mid.sentence =
1369     { "1" }
1370     { "t" }
1371     if$ change.case$
1372     "edition" bibinfo.check
1373     " " * bbl.edition *
1374   }
1375   if$
1376 }

```

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.

```

1377 FUNCTION {format.thesis.type}
1378 { type duplicate$ empty$
1379   'pop$
1380   { swap$ pop$
1381     "t" change.case$ "type" bibinfo.check
1382   }
1383   if$
1384 }

```

`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_□’ (with or without braces, but it should be with), 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.

```

1385 FUNCTION {add.cmd.number}
1386 { ". " swap$ "number" bibinfo.check *
1387 }
1388 FUNCTION {format.legr.number}
1389 { series field.or.null 's :=
1390   s "C" =
1391     'add.cmd.number
1392     { s "Cd" =
1393       'add.cmd.number
1394       { s "Cmd" =
1395         'add.cmd.number
1396         { s "Cmnd" =
1397           'add.cmd.number
1398           { s "Cm" =
1399             'add.cmd.number
1400             { s purify$ #1 #3 substring$ 't :=
1401               t "HL " =
1402                 { ", (" swap$
1403                   "number" bibinfo.check *
1404                   ")" *
1405                 }
1406                 { ", " swap$
1407                   "number" bibinfo.check *
1408                 }
1409               if$
1410             }
1411             if$
1412           }
1413           if$
1414         }
1415         if$
1416       }
1417       if$
1418     }
1419     if$
1420   }
1421 FUNCTION {format.tr.number}
1422 { series duplicate$ empty$
1423   { pop$
1424     type duplicate$ empty$
1425     { pop$
1426       number duplicate$ empty$
1427       { pop$ ""
1428       }
1429       { "(" swap$
1430         "number" bibinfo.check
1431         * ")" *
1432       }
1433       if$
1434     }
1435     { "(" swap$
1436       "t" change.case$ "type" bibinfo.check
1437       *
1438       number duplicate$ empty$
1439       'pop$
1440       { tie.or.space.prefix
1441         "number" bibinfo.check
1442         * *
1443       }
1444       if$
1445       ")" *
1446     }
1447     if$
1448   }
1449   { "t" change.case$ "(" swap$ *
1450     type duplicate$ empty$
1451     { pop$
1452       number duplicate$ empty$

```

```

1453     'pop$
1454     { format.legr.number * }
1455     if$
1456   }
1457   { "t" change.case$ "type" bibinfo.check
1458     *
1459     number duplicate$ empty$
1460     'pop$
1461     { tie.or.space.prefix
1462       "number" bibinfo.check
1463       * *
1464     }
1465     if$
1466   }
1467   if$
1468   ")" *
1469   continue.clause
1470 }
1471 if$
1472 }

```

`format.manual.number` returns the series, type and number fields as one of ‘number’, ‘type number’ ‘series, number’, or ‘series, number. type’. The last of these formulations is for compatibility with previous versions, but may change in future.

```

1473 FUNCTION {format.manual.number}
1474 { series duplicate$ empty$
1475   { pop$
1476     type duplicate$ empty$
1477     { pop$
1478       number duplicate$ empty$
1479       { pop$ ""
1480       }
1481       { "number" bibinfo.check
1482       }
1483       if$
1484     }
1485     { "t" change.case$ "type" bibinfo.check
1486       number duplicate$ empty$
1487       'pop$
1488       { tie.or.space.prefix
1489         "number" bibinfo.check
1490         * *
1491       }
1492       if$
1493     }
1494     if$
1495   }
1496   { "t" change.case$
1497     number duplicate$ empty$
1498     'pop$
1499     { "number" bibinfo.check
1500       ", " swap$ * *
1501     }
1502     if$
1503     type duplicate$ empty$
1504     'pop$
1505     { ". " swap$ *
1506       "t" change.case$ "type" bibinfo.check
1507       *
1508     }
1509     if$
1510   }
1511   if$
1512 }

```

`format.chapter` returns the chapter field in the form ‘c.1’, for use with UK primary legislation.

```

1513 FUNCTION {format.chapter}
1514 { chapter duplicate$ empty$
1515   { pop$ "" }
1516   { "chapter" bibinfo.check

```



```

1517     "c." swap$ *
1518   }
1519   if$
1520 }

```

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

```

1521 FUNCTION {format.series.number.chapter}
1522 { series duplicate$ empty$
1523   { pop$ "(" }
1524   { "series" bibinfo.check
1525     "(" swap$ *
1526   }
1527   if$
1528   type duplicate$ empty$
1529   { pop$
1530     number duplicate$ empty$
1531     'pop$
1532     { "number" bibinfo.check
1533       swap$ duplicate$ "(" =
1534       { swap$ * }
1535       { ", " * swap$ * }
1536     if$
1537   }
1538   if$
1539   chapter duplicate$ empty$
1540   'pop$
1541   { "chapter" bibinfo.check
1542     swap$ duplicate$ "(" =
1543     { swap$ * }
1544     { ", c." * swap$ * }
1545   if$
1546   }
1547   if$
1548 }
1549 { "type" bibinfo.check
1550   swap$ duplicate$ "(" =
1551   { swap$ * }
1552   { ", " * swap$ * }
1553   if$
1554   number duplicate$ empty$
1555   { pop$
1556     chapter duplicate$ empty$
1557     'pop$
1558     { "chapter" bibinfo.check
1559       " c." swap$ * *
1560     }
1561     if$
1562   }
1563   { "number" bibinfo.check
1564     " " swap$ * *
1565     chapter duplicate$ empty$
1566     'pop$
1567     { "chapter" bibinfo.check
1568       ", c." swap$ * *
1569     }
1570     if$
1571   }
1572   if$
1573 }
1574 if$
1575 ")" *
1576 duplicate$ "(" =
1577 { pop$ "" }
1578 'skip$
1579 if$
1580 }

```

`format.case.number` adds to the stack `'(number or casenumber)'` if either of those fields is provided with no institution; or `'(userb or casenumber) Commission Decision number'` if institution is 'Commission'; otherwise adds an empty string.

```

1581 FUNCTION {format.case.number}
1582 { number field.or.null 's :=
1583   s empty$
1584   { casenumber field.or.null 's := }
1585   'skip$
1586   if$
1587   s empty$
1588   { "" }
1589   { institution field.or.null duplicate$ empty$
1590     { pop$
1591       s "number" bibinfo.check
1592       "(" swap$ * ")" *
1593     }
1594     { "Commission" =
1595       { userb field.or.null 's :=
1596         s empty$
1597         { casenumber field.or.null 's := }
1598         'skip$
1599         if$
1600         s duplicate$ empty$
1601         { "" }
1602         { "(" swap$ * ")" * " " * }
1603         if$
1604         number duplicate$ empty$
1605         { pop$ pop$ }
1606         { bbl.commission " " *
1607           swap$ * * *
1608         }
1609         if$
1610       }
1611       { "" }
1612     }
1613   }
1614   if$
1615 }
1616 if$
1617 }

```

case.check checks if the entry should be punctuation with a full stop.

```

1618 FUNCTION {case.check}
1619 { institution duplicate$ empty$
1620   { pop$ continue.clause }
1621   { "Commission" =
1622     'continue.clause
1623     'new.block
1624     if$
1625   }
1626   if$
1627 }

```

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

```

1628 FUNCTION {eu.case.check}
1629 { journal field.or.null "OJ" =
1630   number empty$ not
1631   or
1632   casenumber empty$ not
1633   or
1634   'continue.clause
1635   'new.block
1636   if$
1637 }

```

7.6.5 Notes

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

```

1638 FUNCTION {format.note}
1639 { note empty$
1640   { "" }
1641   { note #1 #1 substring$
1642     duplicate$ "{" =
1643     'skip$
1644     { output.state mid.sentence =
1645       output.state mid.clause =
1646       or
1647       { "l" }
1648       { "u" }
1649       if$
1650       change.case$
1651     }
1652     if$
1653     note #2 global.max$ substring$ * "note" bibinfo.check
1654   }
1655   if$
1656 }

```

7.6.6 Publication block

prepend.address prepends the address field to the string at the top of the stack, which should be a publisher (or equivalent) bibinfo string. If both the input string and address field are non-empty, a colon and space are interposed between them.

```

1657 FUNCTION {prepend.address}
1658 { 't :=
1659   ""
1660   address empty$ t empty$ and
1661   'skip$
1662   { address "address" bibinfo.check *
1663     t empty$
1664     'skip$
1665     { address empty$
1666       'skip$
1667       { ": " * }
1668       if$
1669       t *
1670     }
1671     if$
1672   }
1673   if$
1674 }

```

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

```

1675 FUNCTION {format.publisher.address}
1676 { publisher "publisher" bibinfo.warn prepend.address
1677 }
1678 FUNCTION {format.organization.address}
1679 { publisher empty$
1680   { organization "organization" bibinfo.check prepend.address }
1681   { publisher "publisher" bibinfo.warn prepend.address
1682     organization duplicate$ empty$
1683     'pop$
1684     { "organization" bibinfo.check ". " * swap$ * }
1685     if$
1686   }
1687   if$
1688 }
1689 FUNCTION {format.institution.address}
1690 { publisher empty$
1691   { institution "institution" bibinfo.check prepend.address }
1692   { publisher "publisher" bibinfo.warn prepend.address
1693     institution duplicate$ empty$
1694     'pop$
1695     { "institution" bibinfo.check ". " * swap$ * }
1696     if$
1697   }

```

```
1698 if$
1699 }
```

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

```
1700 FUNCTION {format.library}
1701 { library "library" bibinfo.check
1702   emphasize
```

In `bathx.bst`, we conditionally include the `bbl.online` string:

```
1703 duplicate$ empty$ not
1704 is.online
1705 and
1706 journal empty$
1707 and
1708 { " " * bbl.online *
1709   #1 'online.shown :=
1710 }
1711 'skip$
1712 if$
```

The `library` (or `institution`) 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`.

```
1713 }
1714 FUNCTION {format.img.library}
1715 { library "library" bibinfo.check
1716 }
1717 FUNCTION {format.img.library.address}
1718 { library duplicate$ empty$
1719   { pop$
1720     institution duplicate$ empty$
1721     'skip$
1722     { "institution" bibinfo.check }
1723     if$
1724   }
1725   { "library" bibinfo.check }
1726   if$
1727   duplicate$ empty$
1728   { address "address" bibinfo.check * }
1729   { address duplicate$ empty$
1730     'pop$
1731     { "address" bibinfo.check
1732       swap$ ", " * swap$ *
1733     }
1734     if$
1735   }
1736   if$
1737 }
1738 FUNCTION {format.pub.org.lib.address}
1739 { publisher duplicate$ empty$
1740   { pop$
1741     organization duplicate$ empty$
1742     { pop$
1743       format.img.library.address }
1744     { "organization" bibinfo.check
1745       library empty$
1746       { prepend.address }
1747       { ". " *
1748         format.img.library.address *
1749       }
1750       if$
1751     }
1752     if$
1753   }
1754   { "publisher" bibinfo.warn prepend.address
1755     library empty$
1756     'skip$
```

```

1757     { ". " *
1758       format.img.library *
1759     }
1760   if$
1761 }
1762 if$
1763 }

```

7.6.7 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.

```

1765 INTEGERS { multiresult }
1766 FUNCTION {multi.page.check}
1767 { 't :=
1768   #0 'multiresult :=
1769   { multiresult not
1770     t empty$ not
1771     and
1772   }
1773   { t #1 #1 substring$
1774     duplicate$ "-" =
1775     swap$ duplicate$ "," =
1776     swap$ "+" =
1777     or or
1778     { #1 'multiresult := }
1779     { t #2 global.max$ substring$ 't := }
1780     if$
1781   }
1782   while$
1783   multiresult
1784 }

```

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

```

1785 FUNCTION {n.dashify}
1786 { 't :=
1787   ""
1788   { t empty$ not }
1789   { t #1 #1 substring$ "-" =
1790     { t #1 #2 substring$ "--" = not
1791       { "--" *
1792         t #2 global.max$ substring$ 't :=
1793       }
1794       { { t #1 #1 substring$ "-" = }
1795         { "-" *
1796           t #2 global.max$ substring$ 't :=
1797         }
1798       while$
1799     }
1800     if$
1801   }
1802   { t #1 #1 substring$ *
1803     t #2 global.max$ substring$ 't :=
1804   }
1805   if$
1806 }
1807 while$
1808 }

```

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

```

1809 FUNCTION {format.pages}
1810 { pages duplicate$ empty$ 'skip$
1811   { duplicate$ multi.page.check
1812     { bbl.pages swap$
1813       n.dashify
1814     }
1815     { bbl.page swap$
1816     }

```

```

1817     if$
1818     "pages" bibinfo.check
1819     *
1820   }
1821   if$
1822 }

```

`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.

```

1823 FUNCTION {format.journal.pages}
1824 { pages duplicate$ empty$ 'pop$
1825   { swap$ duplicate$ empty$
1826     { pop$ pop$ format.pages }
1827     { ", " *
1828       swap$
1829       n.dashify
1830       pages multi.page.check
1831       'bbl.pages
1832       'bbl.page
1833       if$
1834       swap$
1835       "pages" bibinfo.check
1836       * *
1837     }
1838     if$
1839   }
1840   if$
1841 }

```

`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).

```

1842 FUNCTION {format.journal.eid}
1843 { eid "eid" bibinfo.check
1844   duplicate$ empty$ 'pop$
1845   { swap$ duplicate$ empty$ 'skip$
1846     { ", " *
1847     }
1848     if$
1849     swap$ *
1850   }
1851   if$
1852 }

```

`format.vol.num` returns volume and number in the form ‘volume(number)’. 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!

```

1853 FUNCTION {format.vol.num}
1854 { volume field.or.null duplicate$ empty$
1855   'skip$
1856   { "volume" bibinfo.check
1857   }
1858   if$
1859   number "number" bibinfo.check duplicate$ empty$
1860   'skip$
1861   { swap$ duplicate$ empty$
1862     { "there's a number but no volume in " cite$ * warning$ }
1863     'skip$
1864     if$
1865     swap$
1866     "(" swap$ * ")" *
1867   }
1868   if$
1869 }

```

`format.vol.num.pages` returns the journal pinpointing in the form ‘volume(number), eid/

pages’. Since the Harvard (Bath) style places publication state information in the volume position, the `pubstate` field from `biblatex` is emulated here. The `eventyear` if present is used instead of month, in order to support parliamentary debates.

```

1870 FUNCTION {format.vol.num.pages}
1871 { pubstate field.or.null
1872   duplicate$ "inpress" =
1873     { pop$ bbl.inpress }
1874     { duplicate$ "submitted" =
1875       { pop$ bbl.submitted }
1876       { duplicate$ "inpreparation" =
1877         { pop$ bbl.inpreparation }
1878         { pop$ "" }
1879       if$
1880     }
1881   if$
1882 }
1883 if$
1884 duplicate$ empty$
1885 { pop$
1886   format.vol.num *
1887   eventyear field.or.null duplicate$ empty$
1888   { pop$
1889     month "month" bibinfo.check duplicate$ empty$
1890     'pop$
1891     { swap$ duplicate$ empty$
1892       'pop$
1893       { ", " * swap$ * }
1894     if$
1895   }
1896   if$
1897 }
1898 { swap$ duplicate$ empty$
1899   'pop$
1900   { ", " * swap$ * }
1901   if$
1902 }
1903 if$
1904 eid empty$
1905 { format.journal.pages }
1906 { format.journal.eid }
1907 if$
1908 }
1909 'skip$
1910 if$
1911 }

```

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

```

1912 FUNCTION {format.chapter.pages}
1913 { chapter empty$
1914   'format.pages
1915   { type empty$
1916     { bbl.chapter }
1917     { type "1" change.case$
1918       "type" bibinfo.check
1919     }
1920   if$
1921   chapter tie.or.space.prefix
1922   "chapter" bibinfo.check
1923   * *
1924   pages empty$
1925   'skip$
1926   { ", " * format.pages * }
1927   if$
1928 }
1929 if$
1930 }

```

`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.

```
1932 FUNCTION {format.journal.series.vol.pages}
1933 { journal field.or.null
1934   duplicate$ "OJ" =
1935     { "journal" bibinfo.check
1936       emphasize
1937       series empty$
1938       volume empty$
1939       and
1940       pages empty$
1941       and
1942       'skip$
1943       { " " * }
1944     if$
1945     format.series
1946     volume "volume" bibinfo.check *
1947     duplicate$ empty$
1948     'skip$
1949     { pages field.or.null duplicate$ empty$
1950       'skip$
1951       { "/" swap$ *
1952       }
1953       if$ *
1954     }
1955     if$ *
1956   }
1957   { duplicate$ "ECR" =
1958     { "journal" bibinfo.check
1959       volume "volume" bibinfo.check
1960       duplicate$ empty$
1961       'skip$
1962       { " " swap$ * "--" * * }
1963     if$
1964   }
1965   { "journal" bibinfo.check
```

In bathx.bst, we conditionally include the bbl.online string:

```
1966   is.online
1967   { " " * bbl.online *
1968     #1 'online.shown :=
1969   }
1970   'skip$
1971   if$
```

```
1972   volume "volume" bibinfo.check
1973   duplicate$ empty$
1974   'pop$
1975   { " " * swap$ * }
1976   if$
1977   pages empty$
1978   eid empty$
1979   and
1980   'skip$
1981   { " " * }
1982   if$
1983   }
1984   if$
1985   }
1986   if$
1987   eid empty$
1988   { pages "pages" bibinfo.check * }
1989   { eid "eid" bibinfo.check * }
1990   if$
1991 }
```

If institution is given (not 'Commission'), format.inst.journal.series.vol.pages adds the institution, number and regular journal information to the stack. otherwise, it calls format.journal.series.vol.pages instead.


```

1992 FUNCTION {format.inst.journal.series.vol.pages}
1993 { institution duplicate$ empty$
1994   { pop$ format.journal.series.vol.pages }
1995   { duplicate$ "Commission" =
1996     { pop$ format.journal.series.vol.pages }
1997     { number duplicate$ empty$
1998       { pop$ casenumber duplicate$ empty$
1999         'skip$
2000         { "casenumber" bibinfo.check
2001           ", " swap$ * *
2002         }
2003         if$
2004       }
2005       { "number" bibinfo.check
2006         ", " swap$ * *
2007       }
2008     } if$
2009     output new.sentence
2010     journal "t" change.case$
2011     "journal" bibinfo.check
2012     emphasize output

```

In `bathx.bst`, we conditionally include the `bbl.online` string:

```

2013 is.online
2014   { continue.clause
2015     bbl.online output
2016     #1 'online.shown :=
2017   }
2018   'skip$
2019   if$

```

```

2020 volume field.or.null
2021 duplicate$ empty$ 'skip$
2022   { "volume" bibinfo.check
2023   }
2024   if$
2025 eventyear field.or.null duplicate$ empty$
2026   { pop$
2027     month "month" bibinfo.check duplicate$ empty$
2028     'pop$
2029     { swap$ duplicate$ empty$
2030       'pop$
2031       { ", " * swap$ * }
2032     } if$
2033   }
2034   if$
2035   }
2036   { swap$ duplicate$ empty$
2037   'pop$
2038   { ", " * swap$ * }
2039   if$
2040   }
2041   if$
2042   eid empty$
2043   { format.journal.pages }
2044   { format.journal.eid }
2045   if$
2046   }
2047   if$
2048   }
2049   if$
2050 }

```

7.6.8 Access and URLs

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

```

2051 FUNCTION {format.eprint}
2052 { eprint duplicate$ empty$
2053   'skip$

```

```

2054 { "\eprint"
2055   archive empty$
2056   'skip$
2057   { "[" * archive * "]" * }
2058   if$
2059   "{" * swap$ * "}" *
2060 }
2061 if$
2062 }

```

`format.url` returns a string Available from: `\url{url}` [Accessed `urldate`]. We also allow the `library` field to be used as a fallback URL.

```

2063 FUNCTION {format.url}
2064 { doi empty$
2065   { url }
2066   { "https://doi.org/" doi * }
2067   if$
2068   duplicate$ empty$
2069   { pop$ "" }
2070   { "\urlprefix\url{" swap$ * "}" * }
2071   if$
2072   duplicate$ empty$
2073   urlyear empty$ not
2074   urldate empty$ not
2075   or
2076   and
2077   { library duplicate$ empty$
2078     'pop$
2079     { "\urlprefix{" swap$ * * }
2080     if$
2081     }
2082     'skip$
2083     if$
2084     urlyear empty$
2085     { urldate empty$
2086       'skip$
2087       { " [\urldateprefix{" * urldate * "}" * }
2088       if$
2089     }
2090     { " [\urldateprefix{" * urlyear * "}" * }
2091     if$
2092   }

```

7.6.9 Related items

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).

```

2093 FUNCTION {format.article.crossref}
2094 { word.in
2095   " \cite{" * crossref * "}" *
2096 }
2097 FUNCTION {format.incoll.inproc.crossref}
2098 { word.in
2099   " \cite{" * crossref * "}" *
2100 }

```

`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.

```

2101 FUNCTION {format.book.crossref}
2102 { volume duplicate$ empty$
2103   { "empty volume in " cite$ * "'s crossref of " * crossref * warning$
2104     pop$ word.in
2105   }
2106   { bbl.volume
2107     capitalize
2108     swap$ tie.or.space.prefix "volume" bibinfo.check * * bbl.of space.word *

```

```

2109     }
2110     if$
2111     " \cite{" * crossref * "}" *
2112 }

```

7.7 Utilities for writing citation data to the file

The start of each entry in the bibliography 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’.

`calc.label` creates the author-year part up to the year, and saves it to the entry’s `label` variable. Note that the closing parenthesis is not included; this is to allow for the `extra.label` to be added if necessary.

```

2114 FUNCTION {calc.label}
2115 { calc.short.authors
2116   short.list
2117   "("
2118   *
2119   year nodate.check duplicate$ empty$
2120     { pop$ "" }
2121     'skip$
2122   if$
2123   *
2124   'label :=
2125 }

```

`output.bibitem` writes out the whole `\bibitem` line, incorporating the label generated by `calc.label`; it supplies the closing parenthesis and conditional inclusion of the full list. The function then starts a new line, puts the entry driver into the `before.all` output state, and puts an empty string at the top of the stack.

```

2126 FUNCTION {output.bibitem}
2127 { newline$
2128   "\bibitem[{" write$
2129   label write$
2130   "]" make.full.names duplicate$ short.list =
2131     { pop$ }
2132     { * }
2133   if$
2134   "]]{" * write$
2135   cite$ write$
2136   "]" write$
2137   newline$
2138   ""
2139   #0 'online.shown :=
2140   before.all 'output.state :=
2141 }

```

7.8 Drivers for formatting specific entry types

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

7.8.1 Article

```

2142 FUNCTION {article}
2143 { output.bibitem
2144   format.authors "author" output.check
2145   author format.key output
2146   format.date "year" output.check
2147   date.block
2148   format.title "title" output.check
2149   print.title.addenda
2150   new.block
2151   crossref missing$
2152     { journal "t" change.case$
2153       "journal" bibinfo.check

```

```

2154     emphasize
2155     "journal" output.check

```

In `bathx.bst`, we conditionally include the `bbl.online` string:

```

2156     is.online
2157     { continue.clause
2158       bbl.online output
2159       #1 'online.shown :=
2160     }
2161     'skip$
2162     if$

```

```

2163     format.vol.num.pages output
2164   }
2165   { format.article.crossref output.nonnull
2166     format.pages output
2167   }
2168   if$
2169   new.block
2170   format.note output
2171   new.block
2172   format.eprint output
2173   format.url output
2174   fin.entry
2175 }

```

7.8.2 Book

```

2176 FUNCTION {book}
2177 { output.bibitem
2178   author empty$
2179   { editor empty$
2180     type$ "reference" =
2181     or
2182     { type$ "reference" =
2183       { format.title "title" output.check }
2184       { format.btitle "title" output.check }
2185     if$
2186     print.labeltitle.addenda
2187     format.date "year" output.check
2188     year empty$ 'skip$ { date.block } if$
2189     print.title.addenda
2190   }
2191   { format.editors "author and editor" output.check
2192     format.date "year" output.check
2193     date.block
2194     format.btitle "title" output.check
2195     print.title.addenda
2196   }
2197   if$
2198 }
2199 { format.authors output.nonnull
2200   crossref missing$
2201   { "author and editor" editor either.or.check }
2202   'skip$
2203   if$
2204   format.date "year" output.check
2205   date.block
2206   format.btitle "title" output.check
2207   print.title.addenda
2208 }
2209 if$
2210 new.block
2211 format.edition output
2212 edition empty$
2213 { mid.sentence 'output.state := }
2214 'new.block
2215 if$
2216 format.bvolume output
2217 format.number.series output

```

```

2218 new.block
2219 format.note output
2220 crossref missing$
2221   { keywords field.or.null "uksi" =
2222     { mid.sentence 'output.state := }
2223     { new.sentence }
2224   if$
2225     format.publisher.address output
2226   }
2227   { new.block
2228     format.book.crossref output.nonnull
2229   }
2230 if$
2231 new.block
2232 format.eprint output
2233 format.url output
2234 fin.entry
2235 }

```

7.8.3 Unpublished or self-published monograph

```

2236 FUNCTION {booklet}
2237 { output.bibitem
2238   author empty$
2239   { format.avtitle "title" output.check
2240     print.labeltitle.addenda
2241     format.date "year" output.check
2242     year empty$ 'skip$ { date.block } if$
2243     print.title.addenda
2244   }
2245   { format.authors output
2246     author format.key output
2247     format.date "year" output.check
2248     date.block
2249     format.avtitle "title" output.check
2250     print.title.addenda
2251   }
2252 if$
2253 new.block
2254 format.series.episode output

```

In bathx.bst, we conditionally include the bbl.online string:

```

2255 is.online
2256   { continue.clause
2257     bbl.online output
2258     #1 'online.shown :=
2259   }
2260 'skip$
2261 if$

```

```

2262 new.block
2263 type "type" bibinfo.check output
2264 new.block
2265 howpublished "howpublished" bibinfo.check output
2266 new.block
2267 format.note output
2268 publisher empty$ 'skip$
2269   { new.block}
2270 if$
2271 format.publisher.address output
2272 new.block
2273 format.eprint output
2274 format.url output
2275 fin.entry
2276 }

```

7.8.4 Work in book

```
2277 FUNCTION {inbook}
2278 { output.bibitem
2279   author empty$
2280   { format.title "title" output.check
2281     print.labeltitle.addenda
2282     format.date "year" output.check
2283     year empty$ 'skip$ { date.block } if$
2284     print.title.addenda
2285   }
2286   { format.authors output.nonnull
2287     format.date "year" output.check
2288     date.block
2289     format.title "title" output.check
2290     print.title.addenda
2291   }
2292   if$
2293   new.block
2294   crossref missing$
2295     { format.in.ed.booktitle "booktitle" output.check
2296       new.block
2297       format.edition output
2298       edition empty$
2299         { mid.sentence 'output.state := }
2300         'new.block
2301       if$
2302       format.bvolume output
2303       format.number.series output
2304       new.block
2305       format.note output
2306       new.block
2307       format.publisher.address output
2308       format.chapter.pages output
2309     }
2310     { format.note output
2311       new.block
2312       format.incoll.inproc.crossref output.nonnull
2313       format.chapter.pages output
2314     }
2315     if$
2316     new.block
2317     format.eprint output
2318     format.url output
2319     fin.entry
2320 }
```

7.8.5 Work in collection

```
2321 FUNCTION {incollection}
2322 { output.bibitem
2323   author empty$
2324     { type$ "inreference" =
2325       { format.label.booktitle output}
2326       { format.title "title" output.check
2327         print.labeltitle.addenda
2328       }
2329       if$
2330       format.date "year" output.check
2331       year empty$ 'skip$ { date.block } if$
2332       type$ "inreference" =
2333       'skip$
2334       { print.title.addenda }
2335       if$
2336     }
2337     { format.authors output.nonnull
2338       format.date "year" output.check
2339       date.block
2340       format.title "title" output.check
2341       print.title.addenda
2342     }
2343     if$
```

```

2344 new.block
2345 crossref missing$
2346 { type$ "inreference" =
2347   author empty$
2348   and
2349     { format.byeditors output }
2350     { format.in.ed.booktitle "booktitle" output.check }
2351   if$
2352     new.block
2353     format.edition output
2354     type$ "inreference" =
2355     author empty$
2356     and
2357       { new.block
2358         format.btitle "title" output.check
2359         print.title.addenda
2360         new.block
2361       }
2362       { edition empty$
2363         { mid.sentence 'output.state := }
2364         'new.block
2365         if$
2366       }
2367     if$
2368     format.bvolume output
2369     format.number.series output
2370     new.block
2371     format.note output
2372     new.block
2373     format.publisher.address output
2374     format.chapter.pages output
2375   }
2376 { format.note output
2377   new.block
2378   format.incoll.inproc.crossref output.nonnull
2379   format.chapter.pages output
2380 }
2381 if$
2382 new.block
2383 format.eprint output
2384 format.url output
2385 fin.entry
2386 }

```

7.8.6 Conference paper

```

2387 FUNCTION {inproceedings}
2388 { output.bibitem
2389   format.authors "author" output.check
2390   author format.key output
2391   format.date "year" output.check
2392   date.block
2393   format.title "title" output.check
2394   print.title.addenda
2395   new.block
2396   crossref missing$
2397     { format.in.ed.booktitle "booktitle" output.check
2398       format.bvolume output
2399       format.number.series output
2400       eventyear output
2401       venue output
2402       format.note output
2403       new.sentence
2404       publisher empty$
2405         { format.organization.address output }
2406         { organization "organization" bibinfo.check output
2407           format.publisher.address output
2408         }
2409       if$
2410       format.pages output
2411     }
2412   { format.note output

```

```

2413     new.block
2414     format.incoll.inproc.crossref output.nonnull
2415     format.pages output
2416   }
2417   if$
2418   new.block
2419   format.eprint output
2420   format.url output
2421   fin.entry
2422 }
2423 FUNCTION {conference} { inproceedings }

```

7.8.7 Legal case studies

```

2424 FUNCTION {jurisdiction}
2425 { output.bibitem
2426   author empty$
2427   { format.btitle "title" output.check
2428     continue.clause
2429     format.case.number output
2430     continue.clause
2431     print.labeltitle.addenda
2432     continue.clause
2433     format.jur.date "year" output.check
2434     year empty$ 'skip$ { case.check } if$
2435     print.title.addenda
2436   }
2437   { format.authors output
2438     author format.key output
2439     format.jur.date "year" output.check
2440     date.block
2441     continue.clause
2442     format.btitle "title" output.check
2443     print.labeltitle.addenda
2444     print.title.addenda
2445   }
2446   if$
2447   case.check
2448   format.note output
2449   note empty$
2450   'case.check
2451   'new.block
2452   if$
2453   crossref missing$
2454   { continue.clause
2455     format.inst.journal.series.vol.pages output
2456   }
2457   { format.article.crossref output.nonnull
2458     format.pages output
2459   }
2460   if$
2461   new.block
2462   format.eprint output
2463   format.url output
2464   fin.entry
2465 }

```

7.8.8 Legislation

```

2466 FUNCTION {uklegislation}
2467 { output.bibitem
2468   author empty$
2469   { format.btitle "title" output.check
2470     print.labeltitle.addenda
2471     continue.clause
2472     format.date emphasize "year" output.check
2473     print.title.addenda
2474   }
2475   { format.authors output
2476     author format.key output
2477     format.date "year" output.check

```



```

2478     date.block
2479     continue.clause
2480     format.btitle "title" output.check
2481     print.labeltitle.addenda
2482     print.title.addenda
2483   }
2484   if$
2485   entrysubtype field.or.null "secondary" =
2486     { number "number" bibinfo.check output }
2487     { series empty$ type empty$ and
2488       { number empty$
2489         { format.chapter output }
2490         { continue.clause
2491           format.series.number.chapter output
2492         }
2493         if$
2494       }
2495       { chapter empty$
2496         { new.block }
2497         { continue.clause }
2498         if$
2499           format.series.number.chapter output
2500       }
2501       if$
2502     }
2503   if$
2504   note empty$
2505   'skip$
2506   { new.block
2507     format.note output
2508     new.block
2509   }
2510   if$
2511   publisher empty$ 'skip$
2512   { entrysubtype field.or.null "secondary" =
2513     'skip$
2514     { new.block }
2515     if$
2516     format.publisher.address output
2517   }
2518   if$
2519   new.block
2520   format.eprint output
2521   format.url output
2522   fin.entry
2523 }
2524 FUNCTION {eulegislation}
2525 { output.bibitem
2526   author empty$
2527   { format.title "title" output.check
2528     print.labeltitle.addenda
2529     continue.clause
2530     format.eu.date "year" output.check
2531     print.title.addenda
2532   }
2533   { format.authors output
2534     author format.key output
2535     format.eu.date "year" output.check
2536     date.block
2537     continue.clause
2538     format.title "title" output.check
2539     print.labeltitle.addenda
2540     print.title.addenda
2541   }
2542   if$
2543   eu.case.check
2544   format.note output
2545   note empty$
2546     'eu.case.check
2547     'new.block
2548   if$
2549   crossref missing$
2550     { continue.clause

```

```

2551     format.journal.series.vol.pages output
2552   }
2553   { format.article.crossref output.nonnull
2554     format.pages output
2555   }
2556   if$
2557   new.block
2558   format.eprint output
2559   format.url output
2560   fin.entry
2561 }
2562 FUNCTION {legislation}
2563 { journal empty$
2564   'uklegislation
2565   'eulegislation
2566   if$
2567 }

```

7.8.9 Parliamentary debate

It would be more efficient to use a straight alias here, but we anticipate using this driver for other templates in future. When that happens, the two sides of the test will be different.

```

2568 FUNCTION {legal}
2569 { entrysubtype field.or.null
2570   "parliamentary" =
2571     'article
2572     'article
2573     if$
2574 }

```

7.8.10 Manual

```

2575 FUNCTION {manual}
2576 { output.bibitem
2577   author empty$
2578     { format.btitle "title" output.check
2579       print.labeltitle.addenda
2580       format.date "year" output.check
2581       year empty$ 'skip$ { date.block } if$
2582       print.title.addenda
2583     }
2584     { format.authors output
2585       author format.key output
2586       format.date "year" output.check
2587       date.block
2588       format.btitle "title" output.check
2589       print.title.addenda
2590     }
2591   if$
2592   month "month" bibinfo.check output
2593   format.edition output
2594   new.block
2595   format.manual.number output
2596   new.block
2597   format.note format.journal.pages output
2598   new.block
2599   format.library output
2600   organization address new.block.checkb
2601   format.organization.address output
2602   new.block
2603   format.eprint output
2604   format.url output
2605   fin.entry
2606 }

```

7.8.11 Image

```
2607 FUNCTION {image}
2608 { output.bibitem
2609   author empty$
2610   { format.btitle "title" output.check
2611     print.labeltitle.addenda
2612     format.date "year" output.check
2613     year empty$ 'skip$ { date.block } if$
2614     print.title.addenda
2615   }
2616   { format.authors output
2617     author format.key output
2618     format.date "year" output.check
2619     date.block
2620     format.btitle "title" output.check
2621     print.title.addenda
2622   }
2623   if$
2624   format.edition output
2625   new.block
2626   format.note output
2627   new.block
2628   format.manual.number output
2629   new.block
2630   format.pub.org.lib.address output
2631   new.block
2632   format.eprint output
2633   format.url output
2634   fin.entry
2635 }
```

7.8.12 Master's thesis

```
2636 FUNCTION {mastersthesis}
2637 { output.bibitem
2638   format.authors "author" output.check
2639   author format.key output
2640   format.date "year" output.check
2641   date.block
2642   format.btitle "title" output.check
2643   print.title.addenda
2644   new.block
2645   format.note output
2646   new.block
2647   bbl.mthesis format.thesis.type output.nonnull
2648   new.block
2649   school "school" bibinfo.warn output
2650   address "address" bibinfo.check output
2651   new.block
2652   format.eprint output
2653   format.url output
2654   fin.entry
2655 }
```

7.8.13 Miscellaneous

```
2656 FUNCTION {misc}
2657 { output.bibitem
2658   author empty$
2659   { format.title "title" output.check
2660     print.labeltitle.addenda
2661     format.date "year" output.check
2662     year empty$ 'skip$ { date.block } if$
2663     print.title.addenda
2664   }
2665   { format.authors output
2666     author format.key output
2667     format.date "year" output.check
2668     date.block
2669     format.title "title" output.check
```

```

2670     print.title.addenda
2671   }
2672   if$
2673   new.block
2674   howpublished "howpublished" bibinfo.check output
2675   new.block
2676   format.note output
2677   new.block
2678   format.eprint output
2679   format.url output
2680   fin.entry
2681 }

```

7.8.14 Doctoral thesis

```

2682 FUNCTION {phdthesis}
2683 { output.bibitem
2684   format.authors "author" output.check
2685   author format.key output
2686   format.date "year" output.check
2687   date.block
2688   format.btitle "title" output.check
2689   print.title.addenda
2690   new.block
2691   format.note output
2692   new.block
2693   bbl.phdthesis format.thesis.type output.nonnull
2694   new.block
2695   school "school" bibinfo.warn output
2696   address "address" bibinfo.check output
2697   new.block
2698   format.eprint output
2699   format.url output
2700   fin.entry
2701 }

```

7.8.15 Conference proceedings

```

2702 FUNCTION {proceedings}
2703 { output.bibitem
2704   editor empty$
2705     { format.btitle "title" output.check
2706       print.labeltitle.addenda
2707       format.date "year" output.check
2708       year empty$ 'skip$ { date.block } if$
2709       print.title.addenda
2710     }
2711     { format.editors output
2712       editor format.key output
2713       format.date "year" output.check
2714       date.block
2715       format.btitle "title" output.check
2716       print.title.addenda
2717     }
2718   if$
2719   format.bvolume output
2720   format.number.series output
2721   eventyear output
2722   venue output
2723   format.note output
2724   new.sentence
2725   publisher empty$
2726     { format.organization.address output }
2727     { organization "organization" bibinfo.check output
2728       format.publisher.address output
2729     }
2730   if$
2731   new.block
2732   format.eprint output
2733   format.url output
2734   fin.entry

```

```
2735 }
```

7.8.16 Report

```
2736 FUNCTION {techreport}
2737 { output.bibitem
2738   author empty$
2739   { format.btitle "title" output.check
2740     print.labeltitle.addenda
2741     format.date "year" output.check
2742     year empty$ 'skip$ { date.block } if$
2743     print.title.addenda
2744   }
2745   { format.authors output
2746     author format.key output
2747     format.date "year" output.check
2748     date.block
2749     format.btitle "title" output.check
2750     print.title.addenda
2751   }
2752   if$
2753   new.block
2754   format.tr.number output
2755   new.block
```

In `bathx.bst`, the `library` field is used for the repository, so the note can go in the same position as in `biblatex-bath`:

```
2756   format.note output
2757   new.block
```

In `bath.bst`, the `note` field is recommended for the repository, therefore has to go after the publisher (unless the `library` field has been used):

```
2609   library empty$
2610   'skip$
2611   { format.note output
2612     new.block
2613   }
2614   if$
```

This bit is the same.

```
2758   publisher empty$
2759   { format.institution.address output }
2760   { institution "institution" bibinfo.check output
2761     format.publisher.address output
2762   }
2763   if$
2764   new.block
```

In `bathx.bst`, the `library` field always comes after the publisher:

```
2765   format.library output
```

In `bath.bst`, we fall back to `note` if `library` is not provided:

```
2622   library empty$
2623   { format.note output }
2624   { format.library output }
2625   if$
```

The rest is the same.

```
2766   new.block
2767   format.eprint output
2768   format.url output
2769   fin.entry
2770 }
```

7.8.17 Unpublished works

```
2771 FUNCTION {unpublished}
2772 { output.bibitem
2773   author empty$
2774   { booktitle empty$ 'format.btitle 'format.title if$ "title" output.check
2775     print.labeltitle.addenda
2776     format.date "year" output.check
2777     year empty$ 'skip$ { date.block } if$
2778     print.title.addenda
2779   }
2780   { format.authors output
2781     author format.key output
2782     format.date "year" output.check
2783     date.block
2784     booktitle empty$ 'format.btitle 'format.title if$ "title" output.check
2785     print.title.addenda
2786   }
2787   if$
2788   new.block
2789   format.in.ed.booktitle output
2790   new.block
2791   howpublished "howpublished" bibinfo.check output
2792   new.block
2793   format.note output
2794   new.block
2795   bbl.unpublished output
2796   new.block
2797   format.eprint output
2798   format.url output
2799   fin.entry
2800 }
```

7.8.18 Aliases and fallbacks

Here are a list of aliases supported by this style.

```
2802 FUNCTION {letter}
2803 { journal empty$
2804   'unpublished
2805   'article
2806   if$
2807 }
2808 FUNCTION {collection} { book }
2809 FUNCTION {reference} { book }
2810 FUNCTION {inreference} { incollection }
2811 FUNCTION {audio} { booklet }
2812 FUNCTION {movie} { booklet }
2813 FUNCTION {music} { booklet }
2814 FUNCTION {video} { booklet }
2815 FUNCTION {dataset} { manual }
2816 FUNCTION {electronic} { manual }
2817 FUNCTION {online} { manual }
2818 FUNCTION {patent} { manual }
2819 FUNCTION {software} { manual }
2820 FUNCTION {standard} { manual }
2821 FUNCTION {www} { manual }
2822 FUNCTION {thesis} { phdthesis }
2823 FUNCTION {report} { techreport }
```

The fallback driver is misc, as with most styles.

```
2824 FUNCTION {default.type} { misc }
```

7.9 Processing the database

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

```
2826 READ
```

7.9.1 Utilities for sorting

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.

```
2827 FUNCTION {sortify}
2828 { purify$
2829   "l" change.case$
2830 }
```

`chop.word` takes a prefix (stop) string, an integer *len*, and a string *s*. If the first *len* characters of *s* match the prefix, those characters are stripped off *s*. We'll use this to strip stop words from the start of titles, for sorting purposes.

```
2831 INTEGERS { len }
2832 FUNCTION {chop.word}
2833 { 's :=
2834   'len :=
2835   s #1 len substring$ =
2836     { s len #1 + global.max$ substring$ }
2837     's
2838   if$
2839 }
```

`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.'

```
2840 FUNCTION {sort.format.names}
2841 { 's :=
2842   #1 'nameptr :=
2843   ""
2844   s num.names$ 'numnames :=
2845   numnames 'namesleft :=
2846   { namesleft #0 > }
2847   { s nameptr
2848     "{ll{ }}{ f{ }}{ jj{ }}{ vv{ }}"
2849     format.name$ 't :=
2850     nameptr #1 >
2851     { " " *
2852       namesleft #1 =
2853       t "others" =
2854       and
2855       { "zzzzz" 't := }
2856       'skip$
2857       if$
2858       t sortify *
2859     }
2860     { t sortify * }
2861     if$
2862     nameptr #1 + 'nameptr :=
2863     namesleft #1 - 'namesleft :=
2864   }
2865   while$
2866 }
```

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

```
2867 FUNCTION {sort.format.title}
2868 { 't :=
2869   "A " #2
2870   "An " #3
2871   "The " #4 t chop.word
2872   chop.word
2873   chop.word
2874   sortify
2875   #1 global.max$ substring$
2876 }
```

`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.

```

2877 FUNCTION {author.editor.sort}
2878 { shortauthor empty$
2879   { author empty$
2880     { shorteditor empty$
2881       { editor empty$
2882         { title empty$
2883           { key empty$
2884             { "to sort, need author, editor, title, or key in "
2885               cite$ * warning$
2886               ""
2887             }
2888             { key sortify }
2889             if$
2890           }
2891           { title sort.format.title }
2892           if$
2893         }
2894         { editor sort.format.names }
2895         if$
2896       }
2897       { shorteditor sort.format.names }
2898       if$
2899     }
2900     { author sort.format.names }
2901     if$
2902   }
2903   { shortauthor sort.format.names }
2904   if$
2905 }

```

`editor.sort` does similar but skips the author names.

```

2906 FUNCTION {editor.sort}
2907 { shorteditor empty$
2908   { editor empty$
2909     { title empty$
2910       { key empty$
2911         { "to sort, need editor, title, or key in " cite$ * warning$
2912           ""
2913         }
2914         { key sortify }
2915         if$
2916       }
2917       { title sort.format.title }
2918       if$
2919     }
2920     { editor sort.format.names }
2921     if$
2922   }
2923   { shorteditor sort.format.names }
2924   if$
2925 }

```

`author.sort` does similar but skips the editor names.

```

2926 FUNCTION {author.sort}
2927 { shortauthor empty$
2928   { author empty$
2929     { type$ "inreference" =
2930       booktitle empty$ not
2931       and
2932       { booktitle sort.format.title }
2933       { title empty$
2934         { key empty$
2935           { "to sort, need author, title, or key in " cite$ * warning$
2936             ""
2937           }
2938           { key sortify }
2939           if$

```



```

2940         }
2941         { title sort.format.title }
2942         if$
2943     }
2944     if$
2945 }
2946 { author sort.format.names }
2947 if$
2948 }
2949 { shortauthor sort.format.names }
2950 if$
2951 }

```

`presort` uses `calc.label` to set the entry's `label` field, then constructs the `sort.label` from it and one of the above three functions. It also constructs the `sort.key$` (used by `SORT`) from the `sort.label` and the title.

```

2952 FUNCTION {presort}
2953 { calc.label
2954   label sortify
2955   " "
2956   *
2957   type$ "book" =
2958   type$ "inbook" =
2959   or
2960   'author.editor.sort
2961   { type$ "proceedings" =
2962     'editor.sort
2963     'author.sort
2964     if$
2965   }
2966   if$
2967   #1 entry.max$ substring$
2968   'sort.label :=
2969   sort.label
2970   *
2971   " "
2972   *
2973   title field.or.null
2974   sort.format.title
2975   *
2976   #1 entry.max$ substring$
2977   'sort.key$ :=
2978 }

```

7.9.2 Sorting and disambiguating entries

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.

```

2980 ITERATE {presort}
2981 SORT

```

We now check to see if the labels we've generated so far are sufficiently distinct.

First of all, we detect if there are several different lead authors (or editors) that share a surname, and if so disambiguate them with an initial. Then we check to see if several different author/editor groups have been truncated to the same lead author/editor (with 'et al.') and if so, try to make the truncated lists more distinct by including more names.

We need a few helper variables for this.

```

2982 INTEGERS { cite.initials.old }
2983 STRINGS { name1f name1f.old name1g name1g.old name.old }

```

The `is.label.source` function takes a field, and returns true if it matches the entry's `label.source` field.

```

2984 FUNCTION {is.label.source}
2985 { duplicate$ empty$

```

```

2986 { pop$ #0 }
2987 { label.source =
2988   { #1 }
2989   { #0 }
2990   if$
2991 }
2992 if$
2993 }

```

The following three functions are conveniences for managing the loop.

```

2994 FUNCTION {initialize.init.stuff}
2995 { "yyyy" 'name1f.old :=
2996   "yyyy" 'name1g.old :=
2997   "yyyy" 'name.old :=
2998   #0 'cite.initials.old :=
2999 }
3000 FUNCTION {end.loop.init}
3001 { name1f 'name1f.old :=
3002   name1g 'name1g.old :=
3003   cite.initials 'cite.initials.old :=
3004 }
3005 FUNCTION {has.no.init}
3006 { duplicate$ duplicate$
3007   'name1f.old :=
3008   'name1g.old :=
3009   'name.old :=
3010   #0 'cite.initials.old :=
3011 }

```

The `cmp.names` function takes the name field used to generate the label, and contrasts it against the value used to generate the previous (or subsequent) entry's label, which should also come from a name field. It determines the 0-based index of the first name that differs between the two lists, and returns whichever is the greater of that number and the entry's current `disamb.names` value. If either list runs out of names before a difference is found, this counts as a 0 index (no need to disambiguate).

```

3012 FUNCTION {cmp.names}
3013 { 's :=
3014   #1 'nameptr :=
3015   disamb.names
3016   s num.names$ 'numnames :=
3017   name.old num.names$ duplicate$ numnames <
3018   { 'numnames := }
3019   'pop$
3020   if$
3021   numnames 'namesleft :=
3022   { namesleft #0 > }
3023   { s nameptr "{1l}" format.name$
3024     name.old nameptr "{1l}" format.name$
3025     =
3026     'skip$
3027     { nameptr #1 -
3028       duplicate$ disamb.names >
3029       { swap$ pop$ }
3030       'pop$
3031       if$
3032       #1 'namesleft :=
3033     }
3034     if$
3035     nameptr #1 + 'nameptr :=
3036     namesleft #1 - 'namesleft :=
3037   }
3038   while$
3039 }

```

With `check.cite.init`, we compare the lead author/editor name with the previous (or subsequent) one. If the surname is different, nothing is done. If the surname is the same but the initial is different, `cite.initials` is set to 1 and the label is regenerated. If both the surname and the initial is the same, `cite.initials` is set to the same as for the previous (subsequent) entry; if the

name list has more than 3 names in it, we additionally run `cmp.names` (above) to set an appropriate value of `disamb.names`; and lastly the label is regenerated.

```

3040 FUNCTION {check.cite.init}
3041 { 's :=
3042   s #1 "{11}" format.name$ 'nameif :=
3043   s #1 "{f.{.}}{ j}{ vv}" format.name$ 'nameig :=
3044   nameif nameif.old =
3045     { nameig nameig.old =
3046       { cite.initials.old 'cite.initials :=
3047         s num.names$ #3 >
3048           { s cmp.names 'disamb.names := }
3049           'skip$
3050         if$
3051         presort
3052       }
3053       { #1 'cite.initials :=
3054         presort
3055         end.loop.init
3056       }
3057     if$
3058   }
3059   { end.loop.init }
3060   if$
3061   s 'name.old :=
3062 }

```

The above functions only need to be run on the name field used to generate the label, so this wrapper function detects whether and which one was.

```

3063 FUNCTION {check.cite.head}
3064 { author duplicate$ is.label.source
3065   { check.cite.init }
3066   { pop$
3067     editor duplicate$ is.label.source
3068     { check.cite.init }
3069     { pop$
3070       shortauthor duplicate$ is.label.source
3071       { check.cite.init }
3072       { pop$
3073         shorteditor duplicate$ is.label.source
3074         { check.cite.init }
3075         { pop$
3076           label.source has.no.init
3077         }
3078       if$
3079     }
3080   if$
3081 }
3082 if$
3083 }
3084 if$
3085 }

```

We run this both forwards and backwards over the list of entries, so each one is compared with both the previous and subsequent entry.

```

3086 EXECUTE {initialize.init.stuff}
3087 ITERATE {check.cite.head}
3088 EXECUTE {initialize.init.stuff}
3089 REVERSE {check.cite.head}
3090 SORT

```

Having disambiguated author/editor lists as much as we can, we now look to see if more than entry has the same head name/title and year. If so, we'll need to add a letter to the year.

We'll need a few helper variables for this.

```

3091 INTEGERS { last.extra.num last.extra.num.extended last.extra.num.blank number.label }
3092 STRINGS { last.label next.extra }

```

The `last.char` function will help us tell if the information in the year slot ends in a number or not.

```

3093 FUNCTION {last.char}
3094 { #1 'len :=
3095   { duplicate$ duplicate$ #1 len substring$ = not }
3096   { len #1 + 'len := }
3097   while$
3098   len #1 substring$
3099 }

```

The following function is a convenience for managing the loop.

```

3100 FUNCTION {initialize.extra.label.stuff}
3101 { #0 int.to.chr$ 'last.label :=
3102   "" 'next.extra :=
3103   #0 'last.extra.num :=
3104   "a" chr.to.int$ #1 - 'last.extra.num.blank :=
3105   last.extra.num.blank 'last.extra.num.extended :=
3106   #0 'number.label :=
3107 }

```

When we go forwards through the entries, we'll set the entry's `extra.label` either to the empty string or a letter. The letter 'b' means the entry shares a label with the previous entry, 'c' means it shares one with the two previous entries, and so on. The first entry with the shared label of course remains unmarked. We add some logic to prepend a non-breaking space to the letter if the ambiguous label didn't end in a numeral (between '/' and ':' in ASCII codes).

```

3108 FUNCTION {forward.pass}
3109 { last.label label =
3110   { last.extra.num #1 + 'last.extra.num :=
3111     last.extra.num "z" chr.to.int$ >
3112     { "a" chr.to.int$ 'last.extra.num :=
3113       last.extra.num.extended #1 + 'last.extra.num.extended :=
3114     }
3115     'skip$
3116     if$
3117     last.extra.num.extended last.extra.num.blank >
3118     { last.extra.num.extended int.to.chr$
3119       last.extra.num int.to.chr$ *
3120     }
3121     { last.extra.num int.to.chr$ }
3122     if$
3123     label last.char chr.to.int$
3124     duplicate$ "/" chr.to.int$ >
3125     swap$ ":" chr.to.int$ < and
3126     'skip$
3127     { "~" swap$ * }
3128     if$
3129     'extra.label :=
3130   }
3131   { "a" chr.to.int$ 'last.extra.num :=
3132     "" 'extra.label :=
3133     label 'last.label :=
3134   }
3135   if$
3136   number.label #1 + 'number.label :=
3137 }

```

When we go backwards through the entries, most of the work has already been done. We just need to watch out for entries with an `extra.label` of 'b', as that means the next one we see will be missing its `extra.label` of 'a'.

```

3138 FUNCTION {reverse.pass}
3139 { next.extra "b" =
3140   { "a" 'extra.label := }
3141   { next.extra "~b" =
3142     { "~a" 'extra.label := }
3143     'skip$
3144     if$

```

```

3145     }
3146     if$
3147     extra.label 'next.extra :=
3148     extra.label
3149     duplicate$ empty$
3150     'skip$
3151     { "\natexlab{" swap$ * "}" * }
3152     if$
3153     'extra.label :=
3154     label extra.label * 'label :=
3155 }

```

Now we complete the forward and backwards passes over the entries.

```

3156 EXECUTE {initialize.extra.label.stuff}
3157 ITERATE {forward.pass}
3158 REVERSE {reverse.pass}

```

Once we have done that, we re-sort the entries to make absolutely sure they in the right order, this time with the year inserted into the `sort.key$` between the `sort.label` and the title.

```

3159 FUNCTION {bib.sort.order}
3160 { sort.label
3161   " "
3162   *
3163   year field.or.null sortify
3164   *
3165   " "
3166   *
3167   title field.or.null
3168   sort.format.title
3169   *
3170   #1 entry.max$ substring$
3171   'sort.key$ :=
3172 }
3173 ITERATE {bib.sort.order}
3174 SORT

```

7.9.3 Writing out the bibliography to a file

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

```

3175 FUNCTION {begin.bib}
3176 { preamble$ empty$
3177   'skip$
3178   { preamble$ write$ newline$ }
3179   if$
3180   "\begin{thebibliography}{ number.label int.to.str$ * "}" *
3181   write$ newline$
3182   "\providecommand{\natexlab}[1]{#1}"
3183   write$ newline$
3184   "\providecommand{\url}[1]{\texttt{#1}}"
3185   write$ newline$
3186   "\providecommand{\urlprefix}{Available from: }"
3187   write$ newline$
3188   "\providecommand{\urldateprefix}{Accessed }"
3189   write$ newline$
3190   "\providecommand{\selectlanguage}[1]{\relax}"
3191   write$ newline$
3192   "\providecommand{\bibinfo}[2]{#2}"
3193   write$ newline$
3194   "\providecommand{\eprint}[2][ ]{\url{#2}}"
3195   write$ newline$
3196 }
3197 EXECUTE {begin.bib}
3198 EXECUTE {init.state.consts}

```

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

```
3199 ITERATE {call.type$}
```

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

```
3200 FUNCTION {end.bib}  
3201 { newline$  
3202   "\end{thebibliography}" write$ newline$  
3203 }  
3204 EXECUTE {end.bib}
```