LaTeX/参考文献管理

维基教科书，自由的教学读本
< LaTeX

For any academic/research writing, incorporating references into a document is an important task. Fortunately, LaTeX has a variety of features that make dealing with references much simpler, including built-in support for citing references. However, a much more powerful and flexible solution is achieved thanks to an auxiliary tool called BibTeX (http://www.bibtex.org) (which comes bundled as standard with LaTeX.)

BibTeX provides for the storage of all references in an external, flat-file database. This database can be linked to any LaTeX document, and citations made to any reference that is contained within the file. This is often more convenient than embedding them at the end of every document written. There is now a centralized bibliography source that can be linked to as many documents as desired (write once, read many!). Of course, bibliographies can be split over as many files as one wishes, so there can be a file containing references concerning *General Relativity* and another about *Quantum Mechanics*. When writing about *Quantum Gravity* (QG), which tries to bridge the gap between these two theories, both of these files can be linked into the document, in addition to references specific to QG.

目录

内嵌系統

引用

BibTeX

總結

Bibliography in the table of contents

Add the Bibliography to the Table of Contents as numbered item

內嵌系統

If you are writing only one or two documents and aren't planning on writing more on the same subject for a long time, maybe you don't want to waste time creating a database of references you are never going to use. In this case you should consider using the basic and simple bibliography support that is embedded within LaTeX.

LaTeX provides an environment called `thebibliography` that you have to use where you want the bibliography; that usually means at the very end of your document, just before the `\end{document}` command. Here is a practical example:

```
\begin{thebibliography}{9}    \bibitem{lamport94}    Leslie    Lamport,
\emph{\LaTeX:   A   Document   Preparation   System}.  Addison   Wesley,
Massachusetts, 2nd Edition, 1994. \end{thebibliography}
```

OK, so what is going on here? The first thing to notice is the establishment of the environment. `thebibliography` is a keyword that LaTeX recognizes as everything between the begin and end tags as being data for the bibliography. The optional argument, which I supplied after the begin statement, is telling LaTeX how wide the item label will be when printed. Note however, that it is not a literal parameter, i.e the number 9 in this case, but a text width. Therefore, I am effectively telling LaTeX that I will only need reference labels of one character in width, which means no more than nine references in total. If you want more than ten, then input a two-digit number, such as '99' which permits fewer than 100 references.

Next is the actual reference entry itself. This is prefixed with the `\bibitem{cite_key}` command. The *cite_key* should be a unique identifier for that particular reference, and is often some sort of mnemonic consisting of any sequence of letters, numbers and punctuation symbols (although not a comma). I often use the surname of the first author, followed by the last two digits of the year (hence *lamport94*). If that author has produced more than one reference for a given year, then I add letters after, 'a', 'b', etc. But, you should do whatever works for you. Everything after the key is the reference itself. You need to type it as you want it to be presented. I have put the different parts of the reference, such as author, title, etc., on different lines for readability. These linebreaks are ignored by LaTeX. I wanted the title to be in italics, so I used the `\emph{}` command to achieve this.

引用

To actually cite a given document is very easy. Go to the point where you want the citation to appear, and use the following: `\cite{cite_key}`, where the *cite_key* is that of the bibitem you wish to cite. When LaTeX processes the document, the citation will be cross-referenced with the bibitems and replaced with the appropriate number citation. The advantage here, once again, is that LaTeX looks after the numbering for you. If it were totally manual, then adding or removing a reference would be a real chore, as you would have to re-number all the citations by hand.

```
Instead of WYSIWYG editors, typesetting systems like \TeX{} or
\LaTeX{} \cite{lamport94} can be used.
```

Referring More Specific

Sometimes you want to refer to a certain page, figure or theorem in a text book. For that you can use the arguments to the `\cite` command:

```
\cite[p.~215]{citation01}
```

The argument, "p. 215", will show up inside the same brackets. Note the tilde in [p.~215], which replaces the end-of-sentence spacing with a non-breakable inter-word space. There are two reasons: end-of-sentence spacing is too wide, and "p." should not be separated from the page number.

Multiple Citations

When a sequence of multiple citations are needed, you should use a single `\cite{}` command. The citations are then separated by commas. Note that you must not use spaces between the citations. Here's an example:

```
\cite{citation01,citation02,citation03}
```

The result will then be shown as citations inside the same brackets.

No Cite

If you only want a reference to appear in the bibliography, but not where it is referenced in the main text, then the `\nocite{}` command can be used, for example:

```
Lamport showed in 1995 something... \nocite{lamport95}.
```

A special version of the command, `\nocite{*}`, includes all entries from the database, whether they are referenced in the document or not.

Natbib

Using the standard LaTeX bibliography support, you will see that each reference is numbered and each citation corresponds to the numbers. The numeric style of citation is quite common in scientific writing. In other disciplines, the author-year style, e.g., (Roberts, 2003), such as *Harvard* is preferred, and is in fact becoming increasingly common within scientific publications. A discussion about which is best will not occur here, but a possible way to get such an output is by the `natbib` package. In fact, it can supersede LaTeX's own citation commands, as Natbib allows the user to easily switch between Harvard or numeric.

The first job is to add the following to your preamble in order to get LaTeX to use the Natbib package:

```
\usepackage{natbib}
```

Natbib commands

Citation command	Output
`\citet{goossens93}`	Goossens et al. (1993)
`\citep{goossens93}`	(Goossens et al., 1993)
`\citet*{goossens93}`	Goossens, Mittlebach, and Samarin (1993)
`\citep*{goossens93}`	(Goossens, Mittlebach, and Samarin, 1993)
`\citeauthor{goossens93}`	Goossens et al.
`\citeauthor*{goossens93}`	Goossens, Mittlebach, and Samarin
`\citeyear{goossens93}`	1993
`\citeyearpar{goossens93}`	(1993)
`\citealt{goossens93}`	Goossens et al. 1993
`\citealp{goossens93}`	Goossens et al., 1993

Also, you need to change the bibliography style file to be used, so edit the appropriate line at the bottom of the file so that it reads: `\bibliographystyle{plainnat}`. Once done, it is basically a matter of altering the existing `\cite` commands to display the type of citation you want.

The main commands simply add a *t* for 'textual' or *p* for 'parenthesized', to the basic `\cite` command. You will also notice how Natbib by default will compress references with three or more authors to the more concise *1st surname et al* version. By adding an asterisk (*), you can override this default and list all authors associated with that citation. There are some other specialized commands that Natbib supports, listed in the table here.

The final area that I wish to cover about Natbib is customizing its citation style. There is a command called `\bibpunct` that can be used to override the defaults and change certain settings. For example, I have put the following in the preamble:

```
\bibpunct{(}{)}{;}{a}{,}{,}
```

The command requires six mandatory parameters.

1. The symbol for the opening bracket.
2. The symbol for the closing bracket.
3. The symbol that appears between multiple citations.
4. This argument takes a letter:
    - *n* - numerical style.
    - *s* - numerical superscript style.
    - *any other letter* - author-year style.
5. The punctuation to appear between the author and the year (in parenthetical case only).
6. The punctuation used between years, in multiple citations when there is a common author. e.g., (Chomsky 1956, 1957). If you want an extra space, then you need `{,~}`.

So as you can see, this package is quite flexible, especially as you can easily switch between different citation styles by changing a single parameter. Do have a look at the Natbib manual (http://www.ctex.org/documents/packages/bibref/natbib.pdf), it's a short document and you can learn even more about how to

Natbib-compatible styles

Style	Source	Description
plainnat	Provided	natbib-compatible version of plain
abbrvnat	Provided	natbib-compatible version of abbrv
unsrtnat	Provided	natbib-compatible version of unsrt
apsrev	REVTeX 4 home page (http://authors.aps.org/revtex4/)	natbib-compatible style for Physical Review journals
rmpaps	REVTeX 4 home page (http://authors.aps.org/revtex4/)	natbib-compatible style for Review of Modern Physics journals
IEEEtranN	TeX Catalogue entry (http://www.ctan.org/tex-archive/help/Catalogue/entries/ieeetran.html)	natbib-compatible style for IEEE publications
achemso	TeX Catalogue entry (http://www.ctan.org/tex-archive/help/Catalogue/entries/achemso.html)	natbib-compatible style for American Chemical Society journals
rsc	TeX Catalogue entry (http://www.ctan.org/tex-archive/help/Catalogue/entries/rsc.html)	natbib-compatible style for Royal Society of Chemistry journals

use it.

BibTeX

I have previously introduced the idea of embedding references at the end of the document, and then using the `\cite` command to cite them within the text. In this tutorial, I want to do a little better than this method, as it's not as flexible as it could be. Which is why I wish to concentrate on using BibTeX.

A BibTeX database is stored as a *.bib* file. It is a plain text file, and so can be viewed and edited easily. The structure of the file is also quite simple. An example of a BibTeX entry:

```
@article{greenwade93,
    author  = "George D. Greenwade",
    title   = "The {C}omprehensive {T}ex {A}rchive {N}etwork ({CTAN})",
    year    = "1993",
    journal = "TUGBoat",
    volume  = "14",
    number  = "3",
    pages   = "342--351"
}
```

Each entry begins with the declaration of the reference type, in the form of `@type`. BibTeX knows of practically all types you can think of, common ones are: *book*, *article*, and for papers presented at conferences, there is *inproceedings*. In this example, I have referred to an article within a journal.

After the type, you must have a left curly brace '`{`' to signify the beginning of the reference attributes. The first one follows immediately after the brace, which is the citation key. This key must be unique for all entries in your bibliography. It is this identifier that you will use within your document to cross-reference it to this entry. It is up to you as to how you wish to label each reference, but there is a loose standard in which you use the author's surname, followed by the year of publication. This is the scheme that I use in this tutorial.

Next, it should be clear that what follows are the relevant fields and data for that particular reference. The field names on the left are BibTeX keywords. They are followed by an equals sign (=) where the value for that field is then placed. BibTeX expects you to explicitly label the beginning and end of each value. I personally use quotation marks ("), however, you also have the option of using curly braces ('{', '}'). But as you will soon see, curly braces have other roles, within attributes, so I prefer not to use them for this job as they can get more confusing. A notable exception is when you want to use characters with umlauts (ü, ö, etc), since their notation is in the format `\"{o}`, and the quotation mark will close the one opening the field, causing an error in the parsing of the reference.

Remember that each attribute must be followed by a comma to delimit one from another. You do not need to add a comma to the last attribute, since the closing brace will tell BibTeX that there are no more attributes for this entry, although you won't get an error if you do.

It can take a while to learn what the reference types are, and what fields each type has available (and which ones are required or optional, etc). So, look at this entry type reference (http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node6.html) and also this field reference (http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node7.html) for descriptions of all the fields. It may be worth bookmarking or printing these pages so that they are easily at hand when you need them.

作者

BibTeX can be quite clever with names of authors. It can accept names in *forename surname* or *surname, forename*. I personally use the former, but remember that the order you input them (or any data within an entry for that matter) is customizable and so you can get BibTeX to manipulate the input and then output it however you like. If you use the *forename surname* method, then you must be careful with a few special names, where there are compound surnames, for example "John von Neumann". In this form, BibTeX assumes that the last word is the surname, and everything before is the forename, plus any middle names. You must therefore manually tell BibTeX to keep the 'von' and 'Neumann' together. This is achieved easily using curly braces. So the final result would be "John {von Neumann}". This is easily avoided with the *surname, forename*, since you have a comma to separate the surname from the forename.

Secondly, there is the issue of how to tell BibTeX when a reference has more than one author. This is very simply done by putting the keyword *and* in between every author. As we can see from another example:

```
@book{goossens93,
    author    = "Michel Goossens and Frank Mittlebach and Alexander Samarin",
    title     = "The LaTeX Companion",
    year      = "1993",
    publisher = "Addison-Wesley",
    address   = "Reading, Massachusetts"
}
```

This book has three authors, and each is separated as described. Of course, when BibTeX processes and outputs this, there will only be an 'and' between the penultimate and last authors, but within the .bib file, it needs the *and*'s so that it can keep track of the individual authors.

標準範例

@article

An article from a magazine or a journal.

- Required fields: author, title, journal, year.
- Optional fields: volume, number, pages, month, note.

```
@article{Xarticle,
    author    = "",
    title     = "",
    journal   = "",
    %volume   = "",
    %number   = "",
    %pages    = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@book

A published book

- Required fields: author/editor, title, publisher, year.
- Optional fields: volume/number, series, address, edition, month, note.

```
@book{Xbook,
    author    = "",
    title     = "",
    publisher = "",
    %volume   = "",
    %number   = "",
    %series   = "",
    %address  = "",
    %edition  = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@booklet

A bound work without a named publisher or sponsor.

- Required fields: title.
- Optional fields: author, howpublished, address, month, year, note.

```
@booklet{Xbooklet,
    %author   = "",
    title     = "",
    %howpublished   = "",
    %address  = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@conference

Equal to inproceedings

- Required fields: author, title, booktitle, year.
- Optional fields: editor, volume/number, series, pages, address, month, organization, publisher, note.

```
@conference{Xconference,
    author    = "",
    title     = "",
    booktitle = "",
    %editor   = "",
    %volume   = "",
    %number   = "",
    %series   = "",
    %pages    = "",
    %address  = "",
    year      = "XXXX",
    %month    = "",
    %publisher= "",
    %note     = "",
}
```

@inbook

A section of a book

- Required fields: author/editor, title, chapter and/or pages, publisher, year.
- Optional fields: volume/number, series, type, address, edition, month, note.

@incollection

A section of a book having its own title.

- Required fields: author, title, booktitle, publisher, year.
- Optional fields: editor, volume/number, series, type, chapter, pages, address, edition, month, note.

@inproceedings

An article in a conference proceedings.

- Required fields: author, title, booktitle, year.
- Optional fields: editor, volume/number, series, pages, address, month, organization, publisher, note.

@manual

Technical manual

- Required fields: title.
- Optional fields: author, organization, address, edition, month, year, note.

@mastersthesis

Master's thesis

- Required fields: author, title, school, year.
- Optional fields: type (eg. "diploma thesis"), address, month, note.

```
@mastersthesis{Xthesis,
    author    = "",
    title     = "",
    school    = "",
    %type     = "diploma thesis",
    %address  = "",
    year      = "XXXX",
    %month    = "",
    %note     = "",
}
```

@misc

Template useful for other kinds of publication

- Required fields: none
- Optional fields: author, title, howpublished, month, year, note.

```
@misc{Xmisc,
    %author     = "",
    %title      = "",
    %howpublished = "",
    %year     = "XXXX",
    %month    = "",
    %note     = "",
}
```

@phdthesis

Ph.D. thesis

- Required fields: author, title, year, school.
- Optional fields: address, month, keywords, note.

@proceedings

The proceedings of a conference.

- Required fields: title, year.
- Optional fields: editor, volume/number, series, address, month, organization, publisher, note.

@techreport

Technical report from educational, commercial or standardization institution.

- Required fields: author, title, institution, year.
- Optional fields: type, number, address, month, note.

```
@techreport{Xtreport,
    author      = "",
    title       = "",
    institution = "",
    %type       = "",
    %number     = "",
    %address    = "",
    year        = "XXXX",
    %month      = "",
    %note       = "",
}
```

@unpublished

An unpublished article, book, thesis, etc.

- Required fields: author, title, note.
- Optional fields: month, year.

非標準範例

@patent

@collection

@electronic

保留大寫字母

In the event that BibTeX has been set to not preserve all capitalization within titles, problems can occur, especially if you are referring to proper nouns, or acronyms. To tell BibTeX to keep them, use the good old curly braces around the letter in question, (or letters, if it's an acronym) and all will be well!

```
title = "The {LaTeX} Companion",
```

Or you can put the whole title in curly braces:

```
    title = "{The LaTeX Companion}",
```

一些其他例子

Below you will find a few additional examples of bibliography entries. The first one covers the case of multiple authors in the Surname, Firstname format, and the second one deals with the incollection case.

```
@article{AbedonHymanThomas2003, author = "Abedon, S. T. and Hyman, P.
and Thomas, C.", year = "2003", title = "Experimental examination of
bacteriophage latent-period evolution as a response to bacterial
availability", journal = "Applied and Environmental Microbiology",
volume = "69", pages = "7499--7506" } @incollection{Abedon1994, author
= "Abedon, S. T.", title = "Lysis and the interaction between free
phages and infected cells", pages = "397--405", booktitle = "Molecular
biology of bacteriophage T4", editor = "Karam, Jim D. Karam and Drake,
John W. and Kreuzer, Kenneth N. and Mosig, Gisela and Hall, Dwight and
Eiserling, Frederick A. and Black, Lindsay W. and Kutter, Elizabeth
and Carlson, Karin and Miller, Eric S. and Spicer, Eleanor", publisher
= "ASM Press, Washington DC", year = "1994" }
```

If you have to cite a website you can use @misc:

```
@MISC{website:fermentas-lambda, AUTHOR = "Fermentas Inc.", TITLE =
"Phage Lambda: description \& restriction map", MONTH = "November",
YEAR = 2008, HOWPUBLISHED =
"\url{http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm}" }
```

The note field comes in handy if you need to add unstructured information, for example that the corresponding issue of the journal has yet to appear:

```
@article{blackholes, author="Bunny, R.", title="Black Holes and Their
Relation to Hiding Eggs", journal="Theoretical Easter Physics",
publisher="Eggs Ltd.", year="2010", note="(to appear)" }
```

Getting current LaTeX document to use your .bib file

At the end of your LaTeX file (that is, after the content, but before `\end{document}`, you need to place the following commands:

```
\bibliographystyle{plain} \bibliography{sample1,sample2,...,samplen} %
Note the lack of whitespace between the commas and the next bib file.
```

Bibliography styles are files recognized by BibTeX that tell it how to format the information stored in the `.bib` file when processed for output. And so the first command listed above is declaring which style file to use. The style file in this instance is `plain.bst` (which comes as standard with BibTeX). You do not need to add the .bst extension when using this command, as it is assumed. Despite its name, the plain style does a pretty good job (look at the output of this tutorial to see what I mean).

The second command is the one that actually specifies the `.bib` file you wish to use. The ones I created for this tutorial were called `sample1.bib`, `sample2.bib`, . . ., `samplen.bib`, but once again, you don't include the file extension. At the moment, the `.bib` file is in the same directory as the LaTeX document too. However, if your .bib file was elsewhere (which makes sense if you intend to maintain a centralized database of references for all your research), you need to specify the path as well, e.g `\bibliography{/some/where/sample}`.

Now that LaTeX and BibTeX know where to look for the appropriate files, actually citing the references is fairly trivial. The \cite{*ref_key*} is the command you need, making sure that the *ref_key* corresponds exactly to one of the entries in the .bib file. If you wish to cite more that one reference at the same time, do the following: \cite{*ref_key1*, *ref_key2*, ..., *ref_keyN*}.

Why won't LaTeX generate any output?

The addition of BibTeX adds extra complexity for the processing of the source to the desired output. This is largely hidden to the user, but because of all the complexity of the referencing of citations from your source LaTeX file to the database entries in another file, you actually need multiple passes to accomplish the task. This means you have to run LaTeX a number of times, where each pass, it will perform a particular task until it has managed to resolve all the citation references. Here's what you need to type:

1. `latex latex_source_code.tex`
2. `bibtex latex_source_code.aux`
3. `latex latex_source_code.tex`
4. `latex latex_source_code.tex`

(Extensions are optional, if you put them note that the bibtex command takes the AUX file as input.)

After the first LaTeX run, you will see errors such as:

```
LaTeX Warning: Citation `lamport94' on page 1 undefined on input line 21.
...
LaTeX Warning: There were undefined references.
```

The next step is to run bibtex on that same LaTeX source (and not on the actual .bib file) to then define all the references within that document. You should see output like the following:

```
This is BibTeX, Version 0.99c (Web2C 7.3.1)
The top-level auxiliary file: latex_source_code.aux
The style file: plain.bst
Database file #1: sample.bib
```

The third step, which is invoking LaTeX for the second time will see more errors like "`LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.`". Don't be alarmed, it's almost complete. As you can guess, all you have to do is follow its instructions, and run LaTeX for the third time, and the document will be output as expected, without further problems.

If you want a pdf output instead of a dvi output you can use `pdflatex` instead of `latex` as follows:

1. `pdflatex latex_source_code.tex`

2. `bibtex latex_source_code.aux`
3. `pdflatex latex_source_code.tex`
4. `pdflatex latex_source_code.tex`

(Extensions are optional, if you put them note that the bibtex command takes the AUX file as input.)

Note that if you are editing your source in vim and attempt to use command mode and the current file shortcut (%) to process the document like this:

1. `:! pdflatex %`
2. `:! bibtex %`

You will get an error similar to this:

1. `I couldn't open file name 'current_file.tex.aux'`

It appears that the file extension is included by default when the current file command (%) is executed. To process your document from within vim, you must explicitly name the file without the file extension for bibtex to work, as is shown below:

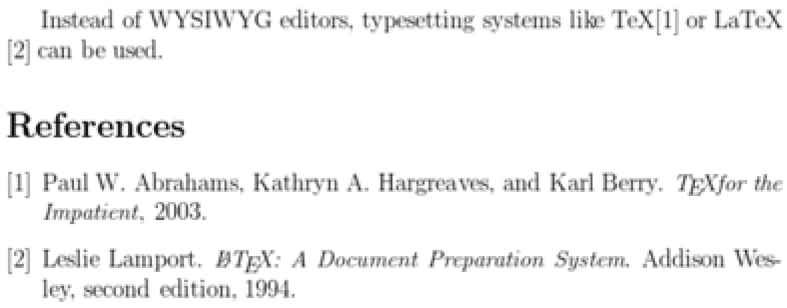
1. `:! pdflatex %`
2. `:! bibtex latex_source_code` (without file extension, it looks for the AUX file as mentioned above)
3. `:! pdflatex %`
4. `:! pdflatex %`

If you are running Unix/Linux or any other platform where you have make (https://en.wikipedia.org/wiki/Make_%28software%29), you can simply create a Makefile and use vim's make command or use make in shell. The Makefile would then look like this:

```
latex_source_code.pdf: latex_source_code.tex latex_source_code.bib

    pdflatex latex_source_code.tex

    bibtex latex_source_code

    pdflatex latex_source_code.tex

    pdflatex latex_source_code.tex
```

参考書目格式

Below you can see three styles available with LaTeX:



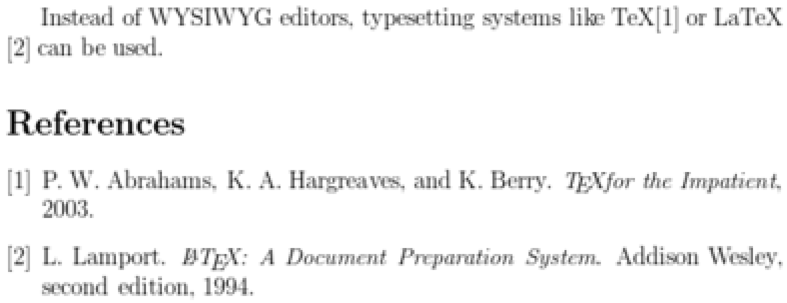
Instead of WYSIWYG editors, typesetting systems like TeX[1] or LaTeX [2] can be used.

References

[1] Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry. *TEXfor the Impatient*, 2003.

[2] Leslie Lamport. *LATEX: A Document Preparation System*. Addison Wesley, second edition, 1994.

plain



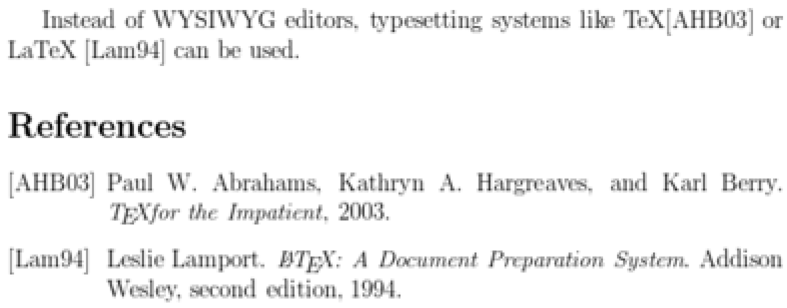
Instead of WYSIWYG editors, typesetting systems like TeX[1] or LaTeX [2] can be used.

References

[1] P. W. Abrahams, K. A. Hargreaves, and K. Berry. *TEXfor the Impatient*, 2003.

[2] L. Lamport. *LATEX: A Document Preparation System*. Addison Wesley, second edition, 1994.

abbrv



Instead of WYSIWYG editors, typesetting systems like TeX[AHB03] or LaTeX [Lam94] can be used.

References

[AHB03] Paul W. Abrahams, Kathryn A. Hargreaves, and Karl Berry. *TEXfor the Impatient*, 2003.

[Lam94] Leslie Lamport. *LATEX: A Document Preparation System*. Addison Wesley, second edition, 1994.

alpha

To number the references in order of appearance, rather than alphabetical order use ieeetr

```
\bibliographystyle{ieeetr}
```

Web page http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html contains more examples.

参考書目含URL連結

As you can see, there is no field for URLs. One possibility is to include Internet addresses in `howpublished` field of `@misc` or `note` field of `@techreport`, `@article`, `@book`: *`howpublished = "\url{http://www.example.com}"`*

Note the usage of `\url` command to ensure proper appearance of URLs.

Another way is to use special field `url` and make bibliography style recognise it.

```
url = "http://www.example.com"
```

You need to use `\usepackage{url}` in the first case or `\usepackage{hyperref}` in the second case.

Styles provided by Natbib (see below) handle this field, other styles can be modified using urlbst (http://purl.org/nxg/dist/urlbst) program. Modifications of three standard styles (plain, abbrv and alpha) are provided with urlbst.

If you need more help about URLs in bibliography, visit FAQ of UK List of TeX (http://www.tex.ac.uk/cgi-bin/texfaq2html?label=citeURL).

自訂參考書目樣貌

One of the main advantages of BibTeX, especially for people who write many research papers, is the ability to customize your bibliography to suit the requirements of a given publication. You will notice how different publications tend to have their own style of formatting references, to which authors must adhere if they want their manuscripts published. In fact, established journals and conference organizers often will have created their own bibliography style (.bst file) for those users of BibTeX, to do all the hard work for you.

It can achieve this because of the nature of the .bib database, where all the information about your references is stored in a structured format, but nothing about style. This is a common theme in LaTeX in general, where it tries as much as possible to keep content and presentation separate.

A bibliography style file (`.bst`) will tell LaTeX how to format each attribute, what order to put them in, what punctuation to use in between particular attributes etc. Unfortunately, creating such a style by hand is not a trivial task. Which is why `Makebst` (also known as *custom-bib*) is the tool we need.

`Makebst` can be used to automatically generate a .bst file based on your needs. It is very simple, and actually asks you a series of questions about your preferences. Once complete, it will then output the appropriate style file for you to use.

It should be installed with the LaTeX distribution (otherwise, you can download it (http://www.mps.mpg.de/software/latex/localtex/localltx.html#makebst)) and it's very simple to initiate. At the command line, type:

```
latex makebst
```

LaTeX will find the relevant file and the questioning process will begin. You will have to answer quite a few (although, note that the default answers are pretty sensible), which means it would be impractical to go through an example in this tutorial. However, it is fairly straight-forward. And if you require further guidance, then there is a comprehensive manual (http://www.mps.mpg.de/software/latex/localtex/doc/merlin.pdf) available. I'd recommend experimenting with it and seeing what the results are when applied to a LaTeX document.

If you are using a custom built .bst file, it is important that LaTeX can find it! So, make sure it's in the same directory as the LaTeX source file, *unless* you are using one of the standard style files (such as *plain* or *plainnat*, that come bundled with LaTeX - these will be automatically found in the directories that they are installed. Also, make sure the name of the .bst file you want to use is reflected in the \bibliographystyle{style} command (but don't include the .bst extension!).

Localizing bibliography appearance

When writing documents in languages other than English, you may find it desirable to adapt the appearance of your bibliography to the document language. This concerns words such as *editors*, *and*, or *in* as well as a proper typographic layout. The babelbib package (http://tug.ctan.org/tex-archive/biblio/bibtex/contrib/babelbib/) can be used here. For example, to layout the bibliography in German, add the following to the header:

```
\usepackage[fixlanguage]{babelbib} \selectbiblanguage{german}
```

Alternatively, you can layout each bibliography entry according to the language of the cited document:

```
\usepackage{babelbib}
```

The language of an entry is specified as an additional field in the BibTeX entry:

```
@article{mueller08,
  % ...
  language = {german}
}
```

For babelbib to take effect, a bibliography style supported by it - one of babplain, babplai3, babalpha, babunsrt, bababbrv, and bababbr3 - must be used:

```
\bibliographystyle{babplain} \bibliography{sample}
```

Getting Bibliographic data

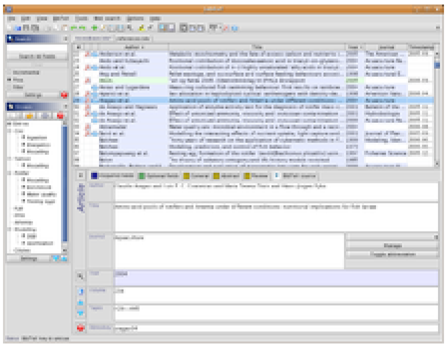
Many online databases provide bibliographic data in BibTeX-Format, making it easy to build your own database. For example, Google Scholar (http://scholar.google.com) offers the option to return properly formatted output, but you must turn it on in the Preferences (http://scholar.google.de/scholar_preferences?hl=en&lr=&output=search).

One should be alert to the fact that bibliographic databases are frequently the product of several generations of automatic processing, and so the resulting BibTex code is prone to a variety of minor errors, especially in older entries.

有用的工具

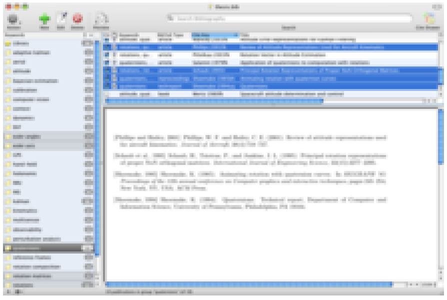
参见：w:en:Comparison of reference management software

- Mendeley (http://mendeley.com) Mendeley is cost-free academic software for managing PDFs which can manage a bibliography in Open Office and read BibTeX.
- Zotero (https://www.zotero.org/) Zotero is a free and open reference manager working as a Firefox plugin capable of importing and exporting bib files.
- Referencer (http://icculus.org/referencer/index.html) Referencer is a Gnome application to organise documents or references, and ultimately generate a BibTeX bibliography file.



JabRef

- Citavi (http://www.citavi.ch) Commercial software (with size-limited free demo version) which even searches libraries for citations and keeps all your knowledge in a database. Export of the database to all kinds of formats is possible. Works together with MS Word and Open Office Writer. Moreover plug ins for browsers and Acrobat Reader exist to automatically include references to your project.
- JabRef (http://jabref.sourceforge.net/) is a small Java program which lets you edit your BibTeX and other bibliographic databases easily, letting you (mostly) forget about the details.
- bibliographer (http://bibliographer.homelinux.net/)(broken link) Bibliographer is a BibTeX bibliography database editor which aims to be easy to use. Its features include linking files to your records with indexing and searching support. The interface is designed for the easy navigation of your bibliography, and double clicking a record will open the linked file.



BibDesk

- cb2Bib (http://www.molspaces.com/cb2bib/) The cb2Bib is a tool for rapidly extracting unformatted, or unstandardized biblographic references from email alerts, journal Web pages, and PDF files.
- KBibTeX (http://www.unix-ag.uni-kl.de/~fischer/kbibtex/) KBibTeX is a BibTeX editor for KDE to edit bibliographies used with LaTeX. Features include comfortable input masks, starting web queries (e. g. Google or PubMed) and exporting to PDF, PostScript, RTF and XML/HTML. As KBibTeX is using KDE's KParts technology, KBibTeX can be embedded into Kile or Konqueror.
- KBib (http://users.tpg.com.au/thachly/kbib/) Another BibTeX editor for KDE. It has similar capabilities, and slightly different UI. Features include BibTeX reference generation from PDF files, plain text, DOI, arXiv & PubMed IDs. Web queries to Google Scholar, PubMer, arXiv and a number of other services are also supported.
- Bibwiki (http://www.plaschg.net/bibwiki/) Bibwiki is a Specialpage for MediaWiki to manage BibTeX bibliographies. It offers a straightforward way to import and export bibliographic records.
- BibDesk (http://bibdesk.sourceforge.net/) BibDesk is a bibliographic reference manager for Mac OS X. It features a very usable user interface and provides a number of features like smart folders based on keywords and live tex display.
- CiteULike (http://www.citeulike.org/) CiteULike is a free online service to organise academic papers. It can export citations in BibTeX format, and can "scrape" BibTeX data from many popular websites.
- Bibtex (http://stat.genopole.cnrs.fr/~cambroise/doku.php?id=softwares:dokuwikibibtexplugin) Bibtex is a DokuWiki plugin that allows for the inclusion of bibtex formatted citations in DokuWiki pages and displays them in APA format. Note: This Plugins is vulnerable to an XSS attack -> http://www.dokuwiki.org/plugin:bibtex
- BibSonomy (http://www.bibsonomy.org/) — A free social bookmark and publication management system based on BibTeX.
- Synapsen (http://www.verzetteln.de/synapsen/) — Hypertextual Card Index / Reference Manager with special support for BiBTeX / biblatex, written in Java.

總結

Although it can take a little time to get to grips with BibTeX, in the long term, it's an efficient way to handle your references. It's not uncommon to find .bib files on websites that people compile as a list of their own publications, or a survey of relevant works within a given topic, etc. Or in those huge, online bibliography databases, you often find BibTeX versions of publications, so it's a quick cut-and-paste into your own .bib file, and then no more hassle!

Having all your references in one place can be a big advantage. And having them in a structured form, that allows customizable output is another one. There are a variety of free utilities that can load your .bib files, and allow you to view them in a more efficient manner, as well as sort them and check for errors.

Bibliography in the table of contents

If you are writing a *book* or *report*, you'll likely insert your bibliography using something like:

```
\begin{thebibliography}{99} \bibitem{bib:one_book} some information
\bibitem{bib:one_article} other information \end{thebibliography}
```

This will create a chapter-like output showing properly all your references. Even though it looks like a chapter, it will not be handled like that so it will not appear on the Table of Contents at the beginning of the document. If you want your bibliography to be in the table of contents, just add the following two lines just before the *thebibliography* environment:

```
\clearpage \addcontentsline{toc}{chapter}{Bibliography}
```

(OR `\addcontentsline{toc}{section}{Bibliography}` if you're writing an *article*)

The first line just terminates the current paragraph and page. If you are writing a *book*, use `\cleardoublepage` to match the style used. The second line will add a line in the Table of Contents (first option, *toc*), it will be like the ones created by chapters (second option, *chapter*), and the third argument will be printed on the corresponding line in the Table of Contents; here *Bibliography* was chosen because it's the same text the *thebibliography* environment will automatically write when you use it, but you are free to write whatever you like. If you are using separate bib file, add these lines between `\bibliographystyle` and `\bibliography`.

If you use hyperref package, you should also use `\phantomsection` command to enable hyperlinking from the table of contents to bibliography.

```
\cleardoublepage    \phantomsection    \addcontentsline{toc}{chapter}
{Bibliography}
```

This trick is particularly useful when you have to insert the bibliography in the Table of Contents, but it can work for anything. When LaTeX finds the code above, it will record the info as described and the current page number, inserting a new line in the Contents page.

Add the Bibliography to the Table of Contents as numbered item

If you instead want bibliography to be numbered section or chapter, you'll likely use this way:

```
\cleardoublepage % This is needed if the book class is used, to place the anchor in the correct
page,
                 % because the bibliography will start on its own page.
                 % Use \clearpage instead if the document class uses the "oneside" argument
\renewcommand*{\refname}{} % This will define heading of bibliography to be empty, so you can...
\section{Bibliography}     % ...place a normal section heading before the bibliography entries.

\begin{thebibliography}{99}
...
\end{thebibliography}
```

Another even easier solution is to use `\section` inside of the `\renewcommand` block:

```
\renewcommand{\refname}{\section{Sources}} % Using "Sources" as the title of the section
\begin{thebibliography}{99}
...
\end{thebibliography}
```

You may wish to use `\renewcommand*{\refname}{\vspace*{-12mm}}` to counteract the extra space the blank `\refname` inserts.

- Note: Use `\bibname` instead of `\refname` if you use the report class.

Template:A-Roberts