Getting to Grips with Latex - Bibliography Management

by Andrew Roberts

0

For any academic/research writing, incorporating your references into your document is an important task. Fortunately, as Latex was aimed for this sort of work, it has a variety of features that make dealing with your references much simpler. Latex has built in support for citing references. However, a much more powerful and flexible solution is achieved thanks to an auxiliary tool called Bibtex (which comes bundled as standard with Latex.)

Bibtex allows you to store all your references in an external, flat-file database. You can then easily link this database to any Latex document, and cite any reference that is contained within the file. This is often more convenient that embedding them at the end of every document you write. You can have a centralised store of your bibliography, that can be linked to as many documents as you wish (write once, read many!) Of course, you can split your bibliographies over as many files as you wish, so you could have a file of references concerning the *theory of relativity*, and another about *quantum physics*. And if you were writing about the *Grand Unifying Theory* (GUT), which tries to bridge the gap between the inconsistencies of these two theories, then you can easily link both to your current document, as well another file of references about GUT, for example. It's up to you how you store your references, of course.

Bibtex

The previous tutorial 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 such as *book*, *article*, and for papers presented at conferences, there is *inproceedings*, etc. 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 with 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 prefer not to use them for this job as they can get more confusing.

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 and also this field reference 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.

Authors

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 customisable 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 in the surname, and everything before is the forename, plus any middle names. You must therefore manually tell Bibtex to keep the 'von' and 'Newman' together. This is achieved easily using curly braces. So the final result would be "John {von Neumann}". If course, this is easily avoided with the *surname, forename*, since you have a comma to separate the surname from the forename. Of course, you use your own preference.

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.

Preserving capital letters

In the event that Bibtex has been set to not preserve all capitalisation within titles, problems can occur, especially if you are referring to proper nouns, or acronyms. To tell Bibtex to keep them, use the good ol' curly braces around the letter in question, (or letters, if its an acronym) and all will be well! As you can see in the following first entry example above:

Getting current Latex document to use your .bib file

This is not actually very difficult. 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{sample}
```

Bibliography styles are files recognised 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 it's 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 one I created for this tutorial was called *sample.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 centralised database of references for all your research), you need to specify the path as well, e.g `\bibliography{$HOME/some/where/sample.bib}`.

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_key3*`}`.

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 bib` (doesn't require .tex extension)
2. `bibtex bib` (doesn't require .bib extension)
3. `latex bib`

4. latex bib

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: bib.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 (as a DVI file. See tutorial 1 if you need reminding how to convert them into PS or PDF.)

What about citation styles other than numerical, e.g., Harvard?

Hopefully, you have already looked at the PDF output of this tutorial. You will have noticed that the document ends with an automatically generated 'References' section. 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. I shall merely provide details for those who wish to have the choice.

Natbib

Natbib is a package written for Latex to do just this job. In fact, it can supersede Latex's own citation commands, as Natbib allows the user to easily switch between Harvard or numeric. In order to demonstrate Natbib, I've modified the original Latex file to take advantage of the additional functionality. Therefore, for the rest of this section, I shall be referring to *bib-har.tex* which can be accessed, along with the PDF of its output, at the bottom of this page.

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

```
\usepackage{natbib}
```

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.

Citation command	Natbib 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)

The main commands simply add a *t* for 'textual' or *p* for 'parenthesised', 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 some other less common commands that Natbib supports, such as `\citeyear`, `\citeauthor`, etc., which are fairly obvious.

The final area that I wish to cover about Natbib is customising 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, it's a short document and you can learn even more about how to use it.

Customising bibliography appearance

In my mind, one the main advantages of Bibtex, especially for people who write many research papers, is the ability to customise 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, which authors must adhere to if they want their manuscript publishing. In fact, established journals and conference organisers 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 - as it should be!

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) 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 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!).

Summary

Although it can take a little time to get to grips with Bibtex, in the long term, it's an efficient way to handle you 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 you references in one place is a big advantage in my opinion. And having then in a structured form, that allows customisable output is another one. There are a variety of free utilities that can load you .bib files, and allow you to view them in a more efficient manner, as well as sorting them and checking for errors.

Files: bib.tex | bib.pdf | bib-har.tex | bib-har.pdf | sample.bib
Useful resources: Natbib manual | Makebst manual | Bibtex entries - A Reference sheet summarising all Bibtex types and fields.

0

Comments? Please leave feedback.

< Back to Latex tutorials.
<< Back to misc.

Last modified: ???

^ ^Top