

Bibliographies

Managing bibliographies with L^AT_EX

Lapo F. Mori

Abstract

The bibliography is a fundamental part of most scientific publications. This article presents and analyzes the main tools that L^AT_EX offers to create, manage, and customize both the references in the text and the list of references at the end of the document.

1 Introduction

Bibliographic references are an important, sometimes fundamental, part of academic documents. In the past, preparation of a bibliography was difficult and tedious mainly because the entries were numbered and ordered by hand. L^AT_EX, which was developed with this kind of document in mind, provides many tools to automatically manage the bibliography and make the authors' work easier. How to create a bibliography with L^AT_EX is described in section 2, starting from the basics and arriving at advanced customization. Bibliographic styles for both the list of references and in-text-citations are analyzed in section 3. The last two sections (4 and 5) analyze two of the most powerful packages available: natbib and BIBL^AT_EX.

2 Bibliography with L^AT_EX

There are two main ways to compose a bibliography with L^AT_EX: automatically with the BIBL^AT_EX program that uses external bibliographic databases (section 2.2), or manually with the thebibliography environment that allows to include all the bibliographic information in the source .tex file (section 2.3). Regardless from the strategy chosen, citations can be added to the text with the same commands, as shown in section 2.1.

2.1 References in the text

Citations can be added to the text with the command `\cite{key}` (and its variants), where `key` corresponds to the citekey field in the .bib file (if using BIBL^AT_EX, section 2.2) or the key of `\bibitem` (if using the thebibliography environment, section 2.3). When compiling the source, `\cite{key}` is linked to the respective `\bibitem` and substituted by the appropriate reference (numbered, author-year, or footnote depending on the style chosen).

Multiple citations can be added by separating with a comma the bibliographic keys inside the same `\cite` command; for example

```
\cite{Goossens1995,Kopka1995}
```

gives

```
(Goossens et al., 1995; Kopka and Daly, 1995)
```

Bibliographic entries that are not cited in the text can be added to the bibliography with the `\nocite{key}` command. The `\nocite{*}` command adds all entries to the bibliography.

2.2 Automatic creation with BIBL^AT_EX

BIBL^AT_EX is a separate program from L^AT_EX that allows creating a bibliography from an external database (.bib file). These databases can be conveniently shared by different L^AT_EX documents. BIBL^AT_EX, which will be described in the following paragraphs, has many advantages over the thebibliography environment; in particular, automatic formatting and ordering of the bibliographic entries.

2.2.1 How BIBL^AT_EX works

BIBL^AT_EX requires:

1. one or more bibliographic databases .bib;
2. a bibliographic style .bst;
3. that the .tex file contains: commands that specify what style .bst and database .bib to use, and citations in the text with the `\cite` and similar commands, as in:

```
\documentclass{...}
...
\begin{document}
See \citet{Kopka1995}.
..
\bibliographystyle{plainnat}
\bibliography{database}
\end{document}
```

4. that the document is compiled in the following order (let's assume that the principal file is called `document.tex`):

```
latex document
bibtex document
latex document
latex document
```

The first time L^AT_EX runs, the `\bibliographystyle` command writes the name of the style .bst to be used into the .aux file, each `\cite` command writes a note into the .aux file, and the `\bibliography` command writes into the .aux file the name of the .bib database(s) to be used. At this stage L^AT_EX does not substitute the `\cite` in the text: the .dvi

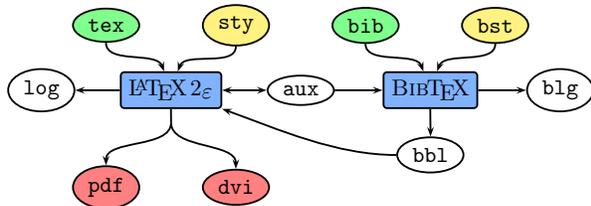


Figure 1: Data flow between \LaTeX and \BibTeX to create a `.pdf` or a `.dvi` (■) from the data files `.tex` and `.bib` (■) and the style files `.sty` and `.bst` (■).

shows question marks for every `\cite` and the `.log` file reports warnings for unknown references.

\BibTeX will then read the `.aux` file, the bibliographic style `.bst` and `.bib` database(s) specified. For every `\cite` entry, it will read the information contained in the `.bib` files and format it according to the style `.bst`. At the end it will order the entries according to the style `.bst` and will write the result into the `.bbl` file. The `.bbl` contains the bibliographic entries organized in a `thebibliography` environment; the entries follow the format specified by the `.bst` style and have the information contained in the `.bib` database. This file will be read every time that the source `.tex` file is compiled.

When running \LaTeX again, it will produce a warning due to the fact that the references are not defined yet and will substitute the `\bibliography` command with the content of the `.bbl` file. It will also write a note into the `.aux` file for every `\bibitem`.

At the next run, \LaTeX will find the definition for every `\cite` in the `.aux` file and all the citations in the text will be resolved. The `.blg` file is a log file created by \BibTeX , similar to the `.log` file for \LaTeX .

This flow, represented in Fig. 1, has to be repeated if the citations in the text, the `.bib` database or the `.bst` style are modified.

2.2.2 Structure of a bibliography database

The `.bib` files are databases that store the information for each bibliographic entry. Here is an example:

```
@BOOK{Kopka1995,
  title = {A Guide to {\LaTeX} -- Document
    Preparation for Beginners and Advanced Users
  },
  publisher = {Addison-Wesley},
  year = {1995},
  author = {H. Kopka and P. W. Daly},
}
```

Each type of entry (`BOOK`, `ARTICLE`, `PROCEEDINGS`, etc.) offers many fields in addition to those shown in the example (`title`, `publisher`, `year`, `author`). Further details can be found in Patashnik (1998).

2.2.3 Rules for using \BibTeX

The syntax of \BibTeX entries is very intuitive. The following paragraphs will discuss the most common rules, all the details can be found in Patashnik (1998).

Capital letters. The \BibTeX styles control capital and lower case letters, especially for the titles. This behavior is very convenient because it ensures a uniform format for the entries but could cause problems in some specific situations such as acronyms, chemical formulæ, etc. In these cases the user needs to enclose in braces the letters whose capitalizations should not be changed by \BibTeX , as in the following example for “CO”:

```
title = {{CO} pollution},
```

The `.bst` style can be used to automatically change the capitalization behavior for titles (e.g. title style or sentence style) instead of manually enclosing all the titles in braces.

Commands. Since \BibTeX modifies the capitalization depending on the `.bst` style, some \LaTeX commands inside titles may not work. For example, if the sentence style is in use and one of the titles contains the command `\LaTeX`, this will be converted to `\latex` and will give the following error:

```
! Undefined control sequence.
```

This also can be avoided by enclosing such commands in braces: `{\LaTeX}`.

Accents and special characters. A similar problem arises with the commands for accents and special characters such as “ö” (`\{o}`), “ç” (`\c{c}`), “ñ” (`\~{n}`), etc. Again, enclosing the command in braces solves the issue:

```
title = {Writing the curriculum vit{\ae}},
```

Name lists. When more than one name is present in the `author` or `editor` fields, they have to be separated by `\and` (a space before and after):

```
author = {Author1 and Author2},
```

Each name has four parts: Name, von, Surname, Jr. The surname is mandatory, all other parts are optional. \BibTeX accepts two different syntaxes for names:

- **Name von Surname:** for example “Pico della Mirandola” has to be written as
Pico della Mirandola.
- **von Surname, Jr, Name:** for example “Pico della Mirandola II” has to be written as
della Mirandola, II, Pico.

The second form is more general because the first one cannot be used when the `Jr` field is present or when the surname contains more than one word and the von part is not present. The `.bst` style takes care of abbreviating (or not) names. If the author wants to use abbreviated names, the `.bib` database can also have abbreviated names. For example:

```
author = {Mori, L. F.},
```

If the list of authors or editors is too long, it can be ended by `and others` that will then be formatted by the style as “et al.”

```
author = {Conte, G. B. and Pianezzolla, E. and
         Chiesa, P. and Rossi, G. and others},
```

URL. The standard `BIBTEX` styles do not provide a field for web addresses. A solution with these styles is to use the `howpublished` field:

```
@MISC{...,
  ...,
  howpublished = {\url{http://...}},
}
```

A better solution is to use styles that provide the `url` field, such as `plainnat` and `abbrevnat` that come with the `natbib` package (section 4) or those that come with the `babelbib` package (section 2.7.3). Moreover, the `url` field is also provided by the custom styles created with `makebst` (section 3.2.2). In all these cases, the syntax is:

```
@ARTICLE{...,
  ...,
  url = {http://...},
}
```

Problems might arise when a web site address is long and close to the write margin. When the document is compiled with `pdfLATEX`, the driver can break the links over several lines. On the other hand, when the document is compiled with the `dvips` driver (`LATEX` → `.dvi` → `PostScript` → `.pdf`), the `breakurl`¹ package must be loaded to support breaking links into several parts while retaining the hyperlink.

Months. `BIBTEX` provides macros to automatically manage the month name specified with the `month` field; these macros automatically manage the full form (March) and abbreviated form (Mar.) and the language (Marzo) depending on the `.bst` style in use. In order to take advantage of these macros, the month has to be written in the abbreviated English form.²

¹ The `breakurl` package was written by Vilar Camara Neto and the last version was released in 2009.

² The abbreviated English form consists in the first three letters of the month: `jan`, `feb`, `mar`, `apr`, `may`, `jun`, `jul`, `aug`, `sep`, `oct`, `nov`, `dec`.

2.2.4 Creating and managing bibliography databases

As should be clear from the example in section 2.2, the `.bib` file syntax is so intuitive that these files can be easily written with any text editor. Several programs, however, are dedicated to the creation and management of `.bib` databases, such as `BibTeXMng`³ (shareware for Windows), `BibDesk`⁴ (open-source for Mac OS X), `KBibTeX`⁵ (open-source for Linux), `Pybliographer`⁶ (open-source for Linux), `bibliographer`⁷ (freeware for Linux), `Bibwiki`⁸ (Internet based), `cb2Bib`⁹ (freeware for Windows and Linux), `Zotero`¹⁰ (open-source multi-platform plugin for Firefox), and `JabRef`¹¹ (multi-platform open-source). The last three programs are particularly interesting and will be discussed in the next paragraphs.

cb2Bib. `cb2bib` (which stands for “clipboard to `BIBTEX`”) is a program for extracting bibliographic information from unformatted sources such as `.pdf` files, web sites, and email. `cb2bib` reads the content of the clipboard and process it according to predefined patterns.¹² If no predefined format pattern is found, `cb2bib` can still be used for manual data extraction.

Zotero. `Zotero` is an open-source multi-platform (Windows, Linux, and Mac OS X) Firefox¹³ plugin that allows to gather, manage, and analyze bibliographic references. Being an extension of a web browser, `Zotero` is particularly useful for extracting bibliographic data from the Internet. The reference database can be exported as a `.pdf` file,¹⁴ a text file, a `BIBTEX` database, and several other formats.

³ <http://www.latexsoft.com/bibtexmng.htm>

⁴ <http://bibdesk.sourceforge.net/>

⁵ <http://www.unix-ag.uni-kl.de/~fischer/kbibtex/>

⁶ <http://www.pybliographer.org/Welcome/>

⁷ <http://bibliographer.homelinux.net/>

⁸ <http://wolfgang.plaschg.net/bibwiki/>

⁹ http://www.molspaces.com/d_cb2bib-overview.php

¹⁰ <http://www.zotero.org/>

¹¹ <http://jabref.sourceforge.net/>

¹² `cb2bib` implements patterns to extract data from the following formats: PNAS Table of Contents Alert (<http://www.pnas.org/>), ISI Web of Knowledge Table of Contents Alert (<http://isiknowledge.com/>), Wiley InterScience Journal Abstracts (<http://www.interscience.wiley.com/>), American Chemical Society Publications (<http://pubs.acs.org/>), ScienceDirect (<http://www.sciencedirect.com/>), Digital Bibliography & Library Project (<http://dblp.uni-trier.de/>), Nature from First Paragraph (<http://www.nature.com/nature/>), IOP Electronic Journals (<http://www.iop.org/EJ/>), JSTOR (<http://www.jstor.org/>), ISI Reference Format (<http://scientific.thomson.com/isi/>), RIS Reference Format.

¹³ <http://www.mozilla.com/firefox/>

¹⁴ The list in the `.pdf` file is formatted according to predefined styles that can be selected by the user.

Zotero can be used together with WYSIWYG¹⁵ software such as Microsoft Word¹⁶ and OpenOffice.¹⁷ Zotero can also perform advanced searches in its libraries and import entries from several formats. A unique feature of Zotero is the ability to create online libraries that can be used from different computers over the Internet.

JabRef. JabRef is an open-source multi-platform (Windows, Linux, and Mac OS X) software written in Java for creating and managing bibliographic databases in the BIB_TE_X format. Entries can be created by editor panels whose fields depend on the type of entry (book, article, proceedings, etc.); these panels can also be customized by adding or removing fields.

JabRef can be used to search articles on Medline¹⁸ and Citeseer¹⁹ and to import bibliographic entries from several formats such as BIB_TE_XXML, CSA, Refer/Endnote, ISI Web of Science, SilverPlatter, Medline/Pubmed (XML), Scifinder, OVID, INSPEC, Biblioscape, Sixpack, JSTOR and RIS. JabRef offers advanced search and management based on keywords. Databases can be printed or exported in HTML, Refer/Endnote, Docbook, BIB_TE_XXML, MODS, RTF, and OpenOffice.

JabRef can automatically create the BIB_TE_X keys (for example by taking the first author's surname and the publication year) and insert the citations into several text editors such as LyX, Kile, and WinEdt. JabRef can associate a .pdf file to every entry and open it with external software. It can also associate a url or a DOI;²⁰ in both cases, JabRef can open a web browser on the corresponding page.

Bibliography resources on the Internet. Several web sites, both of journals and of bibliographic databases, can be used to export entries directly in the BIB_TE_X format. Some of the journals are ACM,²¹ Science,²² PNAS,²³ and The Journal of Chemical Physics.²⁴ Some of the databases are Lead2Amazon²⁵ (a web site that uses Amazon.com, .ca, .co.uk, .de, .fr, .co.jp to automatically generate BIB_TE_X entries),

Google Scholar²⁶ (Google search engine dedicated to scientific publications; select “BIB_TE_X” in “Scholar Preferences”), BibSonomy²⁷ (web site to share links to publications), CiteSeer²⁸ (search engine and digital library for scientific articles), CiteULike²⁹ (web site to share links to publications), The Collection of Computer Science Bibliographies³⁰ (computer science bibliographic database with more than two million entries), HubMed³¹ (alternative interface to PubMed³² that can export entries in BIB_TE_X format), T_EXMed³³ (another alternative interface to PubMed that can export entries in BIB_TE_X).

2.3 Manual composition: thebibliography

The thebibliography environment must be placed in the source document where the author wants the list of references to appear, typically just before `\end{document}`.

```
\documentclass{...}
\begin{document}
...
\begin{thebibliography}{argument}
...
\end{thebibliography}
\end{document}
```

The argument of thebibliography defines the maximum length of the labels. If the labels are automatically generated by L^AT_EX, usually the argument is chosen as “9”, if the list contains less than ten entries, “99”, if less than one hundred, etc. When using customized labels (e.g.: [Mori 06]) the argument must be the widest label.

The thebibliography environment works in a very similar way to the itemize environment. Each entry of the list begins with the `\bibitem` command and its argument, that works as reference identifier (similarly to `\label`), followed by the information about the entry (e.g. author, title, editor, year of publication), with explicit formatting and punctuation. The following example

```
\begin{thebibliography}{9}
\bibitem{Kopka1995} \textsc{Kopka}, H. and \textsc{Daly}, P.~W. (1995). \emph{A Guide to {\LaTeX} --- Document Preparation for Beginners and Advanced Users}. Addison-Wesley.
\end{thebibliography}
```

produces Fig. 2.

¹⁵ Acronym for “What You See Is What You Get”.

¹⁶ <http://office.microsoft.com/>

¹⁷ <http://www.openoffice.org/>

¹⁸ <http://www.ncbi.nlm.nih.gov/sites/entrez>

¹⁹ <http://citeseer.ist.psu.edu/>

²⁰ DOI is the acronym for Digital Object Identifier and represents the future for references to electronic publications. Unlike an url, a DOI is associated with an object (scientific article) and not to the place where it is stored (web site). This guarantees a longer longevity of the link. More information can be found at <http://www.doi.org/>.

²¹ <http://portal.acm.org/>

²² <http://www.sciencemag.org/>

²³ <http://www.pnas.org/>

²⁴ <http://jcp.aip.org/>

²⁵ <http://keijisaito.info/lead2amazon/e/>

²⁶ <http://scholar.google.com/>

²⁷ <http://www.bibsonomy.org/>

²⁸ <http://citeseer.ist.psu.edu/>

²⁹ <http://www.citeulike.org/>

³⁰ <http://liinwww.ira.uka.de/bibliography/>

³¹ <http://www.hubmed.org/>

³² <http://pubmed.gov/>

³³ <http://www.sbg.bio.ic.ac.uk/~mueller/TeXMed/>

[1] KOPKA, H. and DALY, P. W. (2003). *A Guide to L^AT_EX—Document Preparation for Beginners and Advanced Users*, Fourth Edition. Addison-Wesley.

Figure 2: List of references obtained with a manual `thebibliography` environment.

2.4 From BIB_TE_X to thebibliography

Some journals require an explicit `thebibliography`. Even in these cases it is possible to manage the bibliography with an external `.bib` database and BIB_TE_X. As a matter of fact, BIB_TE_X just extracts the useful information from the `.bib` database, formats it according to the `.bst` style, and write the output with the `thebibliography` syntax into the `.bbl` file. Hence, at the end of the work, it is possible to copy the content of the `.bbl` file into the `.tex` file.

2.5 From thebibliography to BIB_TE_X

There is no automatic method to convert the content of a `thebibliography` environment into BIB_TE_X format. Often it may be convenient to import entries from online databases. Otherwise `cb2Bib` can be used to try to import the content of a `thebibliography` environment. For both methods refer to section 2.2.4.

2.6 What method to use

BIB_TE_X makes the bibliography management automatic but also has some disadvantages:

- it increases the complexity of the T_EX environment (adding one or more external programs);
- although it is flexible, BIB_TE_X does not allow completely free composition of bibliography entries.

However, the advantages are greater than these disadvantages:

- it automates tedious operations, especially ordering the bibliography entries;
- it automates bibliography formatting;
- several programs are available for creating and managing BIB_TE_X databases.

2.7 Specialties

2.7.1 Renaming the references section

The name of the bibliography section depends on the class and language selected. The `report` and `book` classes use the variable `\bibname` to specify the name; the `babel` package defines this variable as “Bibliography” in English, “Bibliografia” in Italian, and so on. The `article` class uses the variable `\refname`, which becomes “References” in English, “Riferimenti bibliografici” in Italian, and so on.

The `\renewcommand` command can be used to change the names that are assigned by default to the bibliography, as in the following example:

```
\renewcommand\bibname{Useful references}
```

2.7.2 Multiple bibliographies

The `chapterbib`³⁴ package can be used to create separate bibliographies for each file added to the main document with the `\include` command, and not only for each chapter as suggested by its name. The package is very easy to use since the different bibliographies are created automatically by the `\include` commands and do not require special commands for the references in the text.

The `bibunits`³⁵ package can be used to create separate bibliographies for different parts of the document such as chapters, sections, etc. In addition, a global bibliography can be added at the end of the document. This package does not require *ad hoc* commands for the citations in the text, only the `\bibliographyunit{unit}` command in the preamble (where `unit` corresponds to the document structure to be used for the bibliographies, such as `chapter`, `section`, etc.).

The `multibib`³⁶ package can be used to create multiple bibliographies; unlike the `bibunits` and `chapterbib` packages, the bibliographies can be placed anywhere in the document, not only at the end of certain parts. Each bibliography can have a different `.bst` style and `.bib` database. This package, however, requires special commands for the citations in the text: for each bibliography a different type of `cite` command has to be used.

The `multibl`³⁷ package works similarly to `multibib` but does not allow the creation of a general bibliography at the end of the document.

The `bibtopic` and `splitbib` give results similar to `multibib` and `multibl` but follow a different strategy: instead of using special commands for the citations in the text, they require a distinction between the entries of each bibliography. The `bibtopic`³⁸ package requires that each bibliography has a different

³⁴ The `chapterbib` package was written by Donald Arseneau and the last version was released in 2008.

³⁵ The `bibunits` package was written by Thorsten Hansen and the last version was released in 2004.

³⁶ The `multibib` package was written by Thorsten Hansen and the last version was released in 2004.

³⁷ The `multibl` package was written by Apostolos Syropoulos and the last version was released in 2004.

³⁸ The `bibtopic` package was written by Pierre Basso and Stefan Ulrich and the last version was released in 2006.

.bib database. This is particularly convenient when working with database management software such as JabRef that makes very easy to create several small databases from a global one. The `splitbib`³⁹ package, on the contrary, requires that the different categories and the respective entries are defined in the preamble.

2.7.3 Multilingual bibliographies

The `babelbib`⁴⁰ package, together with `babel`, can be used to create multilingual bibliographies in which:

- each entry is in a specific language, or
- all the entries are in the same language.⁴¹

The second case (all the entries are in the same foreign language) can also be handled by creating a style `.bst` with `makebst`, as described in section 3.2.2. For the first case (entries in different languages), `babelbib` is very convenient: the `language` field can be used to specify the language for each entry as in the following example:

```
@BOOK{Lucchesi1989,
  title = {La cucina di Lucchesia e Versilia},
  publisher = {Franco Muzzio Editore},
  year = {1989},
  author = {E. Lucchesi},
  language = {italian},
}
```

3 Bibliography styles

‘Bibliographic style’ can have two meanings:

- the style of the bibliographic entries (usually at the end of the document),
- the style of the references in the text.

The three main styles for the references in the text (numbered, author-year, footnote) are discussed in section 3.1. Even though the style for the references in the text influences the style of the bibliography entries, L^AT_EX separates these two aspects: the style of the bibliography can be controlled with the `.bst` file, as discussed in section 3.2.

3.1 References in the text

There are three main style families for the references in the text: numbered, author-year, and footnote. Each discipline has its own standards that depend on how the bibliography is used (Garcia, 2007).

³⁹ The `splitbib` package was written by Nicolas Markey and the last version was released in 2007.

⁴⁰ The `babelbib` package was written by Harald Harders and the last version was released in 2006.

⁴¹ At present the package only supports Afrikaans, Danish, Dutch, English, Esperanto, Finnish, French, German, Italian, Norwegian, Portuguese, Spanish, and Swedish.

3.1.1 Numbered

Numbered references usually appear in square brackets and use arabic numerals (e.g.: [1]). The main advantage of this type of references is that they can be used both for direct references (e.g.: ‘see [1] as a reference for the theory’) and for indirect ones (e.g.: ‘this has already been shown [1]’). Another advantage is that these references can appear close to parentheses in sentences such as ‘(further details can be found in [1].)’. In general, this form of the reference in the text, as output by `\cite` in L^AT_EX, is independent from the type of sentence in which it appears and this has made this style quite popular.

3.1.2 Author-year

When an article is cited for referring to a theorem or a theory, it is not necessary that the reader knows who wrote it and when. The interested reader can get this information from the reference list at the end of the document. For this reason, the numbered reference style is the most popular for sciences. In the humanities, however, referring to one author rather than another or to one historic period rather than another has significance in and of itself, and it is important that the reader gets this information directly from the text. For this reason, in the humanities the so-called author-year style is preferred. This style summarizes all the relevant information in the reference; usually it reports the first author’s surname and the year of publication, e.g. (Mori et al., 2006).

Since parentheses have already a meaning, this style may lead to misunderstandings. For example it is possible to say ‘... this has already be proven (Mori, 2006)’, but not ‘(Mori, 2006) proved that ...’. For these reasons, the author-year styles provide many variants for the references to solve grammatical or aesthetic problems. Some examples are:

- ‘This has already been proven (Mori, 2006).’
- ‘Further details can be found in Mori (2006).’
- ‘(see [Mori, 2006])’

The author can choose among these variants by using different commands instead of the usual `\cite`.

Packages The most popular packages for author-year citations are `harvard`, `achicago`, and `natbib`. They offer more or less the same features, `natbib` being the most versatile, but their commands follow different logics. The `harvard`⁴² commands are based on the logical function of the reference in the sentence. For

⁴² The `harvard` package was written by Peter Williams and Thorsten Schnier and the last version was released in 1996.

example `\citenoun` has to be used when the reference is a noun. With `achicago`⁴³ the names depend on the form of the reference. For example `\citeA` can be used to have references that contain only the author name (“A” stands for “author”). `natbib`⁴⁴ is based on the same logic of `harvard` and is the most flexible package for managing author-year citations. It will be discussed in section 4.

3.1.3 Footnotes

Some disciplines, mostly in the humanities, use footnote references. This style is especially common in journals that do not have a reference list at the end of each article.

Packages. The `footbib`⁴⁵ package defines the command `\footcite` which formats all references as superscript numbers in square brackets (e.g. ^[1]). The information of each entry is at the bottom of the page and their number does not follow that of the footnotes. The `\footcite` command does not interfere with `\cite`, it is possible to add a list of references at the end of the document.

The `opcit`⁴⁶ package creates entries that are true footnotes and follow their number (e.g. ¹). The package considers which references have already been cited in order to avoid repetitions by automatically using conventional forms such as “Idem” and “op. cit.”. Besides the manual, the interested reader should also read Garcia (2007).

The `jurabib`⁴⁷ package, originally written for German law documents, offers many tools to manage footnote references. Similarly to `opcit`, it formats the bibliographic references as regular footnotes.

The `natbib` package provides the `super` option that, similarly to `footbib`, creates references that do not follow the number of footnotes and are not enclosed in brackets (e.g. ¹). The list of the references appears at the end of the document and not at the bottom of the pages. Although `natbib` offers fewer options for footnote references than the other packages, it uses the same syntax for footnote, author-year, and numbered styles. This allows switching from one style to another by merely changing the package options in the preamble, without changing the references in the text of the document.

⁴³ The `achicago` package was written by Matt Swift and the last version was released in 2001.

⁴⁴ The `natbib` package was written by Patrick Daly and the last version was released in 2009.

⁴⁵ The `footbib` package was written by Eric Domenjoud and the last version was released in 2004.

⁴⁶ The `opcit` package was written by Federico Garcia and the last version was released in 2007.

⁴⁷ The `jurabib` package was written by Jens Berger and the last version was released in 2004.

The `inlinebib`⁴⁸ package can be used for footnote references but is not recommended since it is rather old, does not offer many options, and can be used only with the `indexing.bst` style.

3.1.4 Hybrid approaches

Between author-year and numbered The style `alpha.bst` is halfway between the author-year style, such as ‘(Mori, 2006)’, and the numbered style, such as ‘[1]’. It produces references such as ‘[Mor06]’. Since this style is not as concise as the numbered and not as informative as the author-year, it is not recommended. It is supported by `natbib`.

Between author-year and footnote Some humanities journals report the bibliographic entries directly in the text. The `bibentry`⁴⁹ package can be used for this purpose. It requires only a few modifications to the `.bst` style, hence, it can be used almost with every style.

The `inlinebib` package, created for footnote references, can also be used for in-text-references although it is not recommended, as noted above.

The `jurabib` package, also created for footnote references, can be used for in-text-references with a small number of styles (`jurabib.bst`, `jhuman.bst`, and two styles of the ‘Chicago’ family).

3.2 Style of the reference list

3.2.1 Existing styles

Almost every journal and publisher have their own rules for formatting the bibliography (use of bold-face or italic for the issue or volume number, use of parentheses and of punctuation, etc.). Many journals provide the `BIBTEX` style, hence `TEX` distributions usually come with a lot of bibliographic styles; the Comprehensive `TEX` Archive Network (CTAN)⁵⁰ provides even more styles.

`BIBTEX` comes with four styles (`plain`, `unsrt`, `abbrv` and `alpha`) that were created by the author of the program, Oren Patashnik. These styles, however, do not support the author-year approach (section 3.1.2).

Ken Turner’s web site⁵¹ provides examples of the most popular `.bst` styles. Another excellent review of the available styles is available on the Reed College web site.⁵²

⁴⁸ The `inlinebib` package was written by René Seindal and the last version was released in 1995.

⁴⁹ The `bibentry` package was written by Patrick Daly, also the author of `natbib`, and the last version was released in 2000.

⁵⁰ <http://www.ctan.org/>

⁵¹ <http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html>

⁵² <http://web.reed.edu/cis/help/LaTeX/bibtexstyles.html>

The best way to test a `.bst` style is by using the `xmpl.bib` database that comes with the `BIBTEX` documentation. If for example we want to test the `example.bst` style, we can use the following:

```
\documentclass{article}
\begin{document}
\bibliographystyle{example}
\nocite{*}
\bibliography{xmpl}
\end{document}
```

3.2.2 Customizing the style with `makebst`

There are two main reasons to create custom `.bst` styles:

1. none of the available `.bst` styles satisfy the author (or publisher),
2. the document is written in a language different from English.⁵³

Writing a `.bst` style may prove to be difficult since `BIBTEX` uses a rather non-intuitive programming language. Luckily, Patrick Daly, who is also the author of `natbib` and coauthor of the excellent book about `LATEX` (Kopka and Daly, 2003), wrote a program called `makebst` that can be used to create interactively a customized `.bst` style for `BIBTEX` (and fully compatible with `natbib`). The program is usually distributed as the `custom-bib` package.

The generic style `merlin.mbs` is the heart of the program: it contains alternative code for all aspects of a bibliographic style and is analyzed in detail in Daly (2007b). This file has to be compiled by the `docstrip`⁵⁴ program in order to produce the respective `.bst` style. Since the number of options is very high (about one hundred), the program provides a graphic interface by means of the `makebst.tex` file. The first step consists in compiling `makebst.tex` with either `TEX` or `LATEX`: at this point the user has to answer interactively to the questions that appear on the screen. At the very beginning the user has to select an `.mbs` file, and, depending on the choice, a `docstrip` batch file is created. This batch file can be used to create a bibliographic style with the characteristics of the `.mbs` file: the options that the user can select interactively depend on the chosen `.mbs` file. `merlin.mbs` is a third generation bibliographic style that has replaced `genbst.mbs` (released in November 1993) and its multilingual version `babel.mbs`. Unlike its predecessors, in `merlin.mbs` all the words such as “editor” are represented by variables (in this case `bb1.editor`) that assume different values depending

on the language in use (`bb1.editor` becomes “curatore” in Italian, “editor” in English, “Redakteur” in German, “redacteur” in French, etc.). `merlin.mbs` supports only the options `English` and `babel`; the definitions for all the other languages are provided by separated `.mbs` files (e.g. `italian.mbs`). The language must be chosen at very beginning: when asked “Enter the name of the MASTER file” select `merlin.mbs`, when asked “Name of language definition file” select the `.mbs` file corresponding to the desired language. If the `.mbs` file is not available for the desired language,⁵⁵ the user can select `babel` that, instead of substituting the variables with their translation, substitutes them with commands (in this case `\bbleditor{}`) whose definition must be written in the `babelbst.tex` file.

Except from the language, some of the customizations offered by `merlin.mbs` are:

- author-year or numbered citations;
- criteria for ordering the entries: citation order, year ordered and then by authors, reverse year ordered and then by authors, etc.;
- format for author names: full with surname last, initials and surname, surname and initials, etc.;
- the number of names to report before substituting them with “et al.”;
- formatting for the author names;
- position of the date;
- format for volume, issue, and page number;
- punctuation.

At the end, the procedure creates a `.dbj` file that has to be compiled with `LATEX` in order to obtain the corresponding `.bst` style. If you want to modify a style created with this procedure, it is very convenient to open the `.dbj` file and modify it, instead of answering again to the interactive questions. Further details on `makebst` can be found in Daly (2007a,b).

4 The `natbib` package

The `natbib` package is highly recommended for bibliography customization. The most common options will be described in the following paragraphs and the other details can be found in Daly (2009).

4.1 Compatible styles

`natbib` only works with styles that support its options; the three that come with the package (`plainnat.bst`, `abbrvnat.bst`, and `unsrtnat.bst`) can replace the corresponding `BIBTEX` standard styles (`plain.bst`, `abbrv.bst`, and `unsrt.bst`) with the advantage that

⁵³ Almost all the available styles are in English.

⁵⁴ `docstrip`, written by Frank Mittelbach, is a standard part of `LATEX` distributions.

⁵⁵ At the moment `.mbs` files are available for Catalan, Danish, Dutch, Esperanto, Finnish, French, German, Italian, Norwegian, Polish, Portuguese, Slovene, and Spanish.

Table 1: List of the commands for references in the text and their effect with an author-year style (option `authoryear`).

Reference in the text		
<code>\citet{mori06}</code>	⇒	Mori et al. (2006)
<code>\cite{mori06}</code>	⇒	Mori et al. (2006)
<code>\citet[chap.~2]{mori06}</code>	⇒	Mori et al. (2006, chap. 2)
References with parentheses		
<code>\citep{mori06}</code>	⇒	(Mori et al., 2006)
<code>\citep[chap.~2]{mori06}</code>	⇒	(Mori et al., 2006, chap. 2)
<code>\citep[see][]{mori06}</code>	⇒	(see Mori et al., 2006)
<code>\citep[see][chap.~2]{mori06}</code>	⇒	(see Mori et al., 2006, chap. 2)
References with complete author list		
<code>\citet*{mori06}</code>	⇒	Mori, Lee, and Krishnan (2006)
<code>\citep*{mori06}</code>	⇒	(Mori, Lee, and Krishnan, 2006)
Multiple references		
<code>\citet{mori06,rossi07}</code>	⇒	Mori et al. (2006); Rossi et al. (2007)
<code>\citep{mori06,rossi07}</code>	⇒	(Mori et al., 2006; Rossi et al. 2007)
<code>\citep{mori06,mori07}</code>	⇒	(Mori et al., 2006, 2007)
<code>\citep{mori06a,mori06b}</code>	⇒	(Mori et al., 2006a,b)
References without parentheses		
<code>\citealt{mori06}</code>	⇒	Mori et al. 2006
<code>\citealt*{mori06}</code>	⇒	Mori, Lee e Krishnan 2006
<code>\citealp{mori06}</code>	⇒	Mori et al., 2006
<code>\citealp*{mori06}</code>	⇒	Mori, Lee e Krishnan, 2006
<code>\citealp{mori06,rossi07}</code>	⇒	Mori et al., 2006; Rossi et al., 2007
<code>\citealp[p.~32]{mori06}</code>	⇒	Mori et al., 2006, p. 32

Table 2: List of the commands for references in the text and their effect with a numbered style (option `numbered`).

References in the text		
<code>\citet{mori06}</code>	⇒	Mori et al. [11]
<code>\citet[chap.~2]{mori06}</code>	⇒	Mori et al. [11, chap. 2]
References with parentheses		
<code>\citep{mori06}</code>	⇒	[11]
<code>\cite{mori06}</code>	⇒	[11]
<code>\citep[chap.~2]{mori06}</code>	⇒	[11, chap. 2]
<code>\citep[see][]{mori06}</code>	⇒	[see 11]
<code>\citep[see][chap.~2]{mori06}</code>	⇒	[see 11, chap. 2]
Multiple references		
<code>\citep{mori06a,mori06b}</code>	⇒	[11, 18]

they can be used for both numbered (the only option available for the three original styles) and author-year references. Several other styles that support `natbib` are available on the Internet. This format is also supported by `makebst` (section 3.2.2).

4.2 Commands for in-text references

`natbib` provides two main commands that substitute the regular `\cite` for the citations: `\citet` for the citations in the text and `\citep` for the citations in parentheses. Both of them have a starred version (`\citet*` and `\citep*`) that produce the complete list of authors rather than the abbreviated one. All commands have two optional arguments to add text before and after the reference. Like `\cite`, these commands can be used for multiple citations. The

package also provides commands that remove the parentheses from the citations: `\citealt` instead of `\citet` and `\citealp` instead of `\citep`. Examples of these commands are shown in Tab. 1 for the author-year style (option `authoryear`) and in Tab. 2 for the numbered style (option `number`).

The standard `\cite` command can still be used with `natbib` and it is interpreted as `\citet` for author-year bibliographies (option `authoryear`) and as `\citep` for numbered bibliographies (option `number`).

4.3 Package options

4.3.1 Type of parentheses

References can be enclosed in different types of brackets by using the following options:

- **round** (default): round brackets as in “see Mori (2006)” or “see (2)”;
- **square**: square brackets as in “see Mori [2006]” or “see [2]”;
- **curly**: curly brackets as in “see Mori {2006}” or “see {2}”.

The `\bibpunct` command can also be used to select the bracket type (Daly, 2007b).

4.3.2 Punctuation

The punctuation used to separate multiple references can be selected with the following options:

- **colon** (default): the colon as in “(Mori et al., 2006; Rossi et al., 2007)” or “Mori et al. (2006); Rossi et al. (2007)”;
- **comma**: the comma as in “(Mori et al., 2006, Rossi et al., 2007)” or “Mori et al. (2006), Rossi et al. (2007)”;

`\bibpunct` can also modify the punctuation (Daly, 2007b).

4.3.3 Bibliography style

The bibliographic style can be set by invoking the respective option when loading `natbib`: `authoryear` (default) loads the author-year style, `numbers` the numbered style, and `super` the footnote style. The selected `.bst` style must support the reference type chosen.

4.3.4 Ordering and compressing multiple references

Multiple references such as `\cite{a,b,c,d}` by default produce bad results such as “[2,6,4,3]”. Ordering the references by hand (i.e. `\cite{b,c,d,a}`) would give “[2,3,4,6]” but such manual operations are undesirable. `natbib`, when used with the `numbers` option, provides the `sort&compress` option that automatically orders and compresses multiple references. For example `\cite{a,b,c,d}` would produce “[2–4,6]”.

4.3.5 References from the bibliography to the text

In the bibliography of a long document, it can be useful to provide the page on which each reference appears. Both the `backref` and the `citeref` packages can be used for this purpose, although the former is more modern and can work together with `hyperref`; namely, it can create hyperlinks when used together with `hypernat`. Neither of the two packages can compress the page list (“5, 6, 7” cannot be automatically converted into “5–7”) but they do not repeat a page number if the same reference appears several times in it.

4.3.6 Reducing the space between the references

The bibliography is composed as a list (very similar to `itemize`, `enumerate`, and `description`) and the space between the items can be controlled with the `\itemsep` parameter (UK TUG, 2009):

```
\let\oldbibliography\thebibliography
\renewcommand{\thebibliography}[1]{%
  \oldbibliography{#1}%
  \setlength{\itemsep}{0pt}%
}
```

The `natbib` package offers an even better solution by providing the `\bibsep` parameter that can be used as in the following example:

```
\setlength{\bibsep}{0pt}
```

4.3.7 Style of the numbers in the bibliography

By default, \LaTeX formats the entry numbers in the following way:

```
[1] GARCIA, F. (2007).  $\LaTeX$  and the different
bibliography styles. TUGboat, 28(2).
[2] GOOSSENS, M., MITTELBAACH, F. and
SAMARIN, A. (1995). The  $\LaTeX$  Companion.
Addison-Wesley.
```

This behavior can be customized with commands such as (UK TUG, 2009):

```
\makeatletter
\renewcommand*{\@biblabel}[1]{\hfill#1.}
\makeatother
```

or, if using `natbib`,

```
\renewcommand{\bibnumfmt}[1]{#1.}
```

Both of them give:

```
1. GARCIA, F. (2007).  $\LaTeX$  and the different
bibliography styles. TUGboat, 28(2).
2. GOOSSENS, M., MITTELBAACH, F. e SAMARIN,
A. (1995). The  $\LaTeX$  Companion. Addison-
Wesley.
```

5 BIB \LaTeX

The `BIB \LaTeX` package offers a general solution for customizing bibliographic entries and references. Besides offering features similar to those of the packages analyzed so far, it can be used to modify a bibliographic style by using only \LaTeX commands.

This package, written by Philipp Lehman, is still under development and, although (at the author’s

request) it is not included in many L^AT_EX distributions, it is available on CTAN.⁵⁶ BIBL^AT_EX requires ϵ -T_EX, the etoolbox package, also under development, and the packages keyval, ifthen, and calc. In addition, babel and csquote, while not required, are recommended in order to use the full potential of BIBL^AT_EX. A detailed analysis of BIBL^AT_EX would require a separate article; the following sections only describe its main characteristics. Further details can be found in the package documentation (Lehmann, 2009).

5.1 BIBL^AT_EX styles

The main limitation of BIB_ET_EX is that, in order to have full control of the bibliographic style, the user has to learn a new language that is completely different from that of (L^A)T_EX. The custom-bib package, as explained in section 3.2.2, is very helpful but is not always enough to obtain the desired result.

With BIBL^AT_EX the bibliographic references and citations can be fully controlled with L^AT_EX commands. The style is contained, not in a .bst file, but in a .bbx (bibliographic style) or .cbx (citation style) file. The .bbl file that is created when compiling does not contain a thebibliography environment, but rather a series of macros that contain the bibliographic data.

The style can be specified by declaring a package option:

```
\usepackage[style=numeric]{biblatex}
```

or:

```
\usepackage[bibstyle=authortitle,%
citestyle=verbose-trad1]{biblatex}
```

In the first case the value numeric is assigned both to bibstyle and to citestyle.

The package comes with some styles. The bibliographic styles cover the four traditional categories: numeric, alphabetic, authoryear, and authortitle. A citation style can be associated with each of them to control the references in text (numbered, author-year, footnote). The verbose style uses the complete citation the first time and an abbreviated form, such as *idem*, *ibidem*, *op. cit.*, and *loc. cit.* subsequently.

The user can modify the standard styles inside the document or create new ones. For example, to make the article titles in italic and the journal name enclosed in quotation marks, the following commands can be given in the preamble:

```
\DeclareFieldFormat{article}{title}%
```

⁵⁶ <http://www.ctan.org/tex-archive/macros/latex/expt1/biblatex/>

```
{\mkbibemph{#1\isdot}}
\DeclareFieldFormat{journaltitle}%
{\mkbibquote{#1}}
```

To modify the author-year style in order to use it again with other documents, a user can write the following in the myauthoryear.bbx file:

```
\RequireBibliographyStyle{authoryear}
\DeclareFieldFormat{article}{title}%
{\mkbibemph{#1\isdot}}
\DeclareFieldFormat{journaltitle}%
{\mkbibquote{#1}}
\endinput
```

and specify myauthoryear as the value for bibstyle.

5.2 BIBL^AT_EX commands for references

BIBL^AT_EX, besides the standard \cite and \nocite commands, provides citation commands for different contexts: \parencite encloses the reference in round brackets; \footcite inserts the reference in a footnote; \textcite for when the citation is part of a sentence; \supercite (only for numbered styles) for superscript citations, and \fullcite insert the full bibliographic entry. Finally, \autocite automatically invokes the suitable command of these depending on the context. Examples of these commands are reported in Tab. 3 and 4. Commands to cite parts of the entry are also available: \citeauthor, \citetitle, \citeyear, and \citeurl.

5.3 BIBL^AT_EX commands for bibliographies

The bibliographic list of references in BIBL^AT_EX is inserted differently than with BIB_ET_EX. As discussed earlier, the bibliography style is specified as an option to the package instead of with \bibliographystyle. The \bibliography command only specifies which databases to use, and does not create any list. The entry list can be generated with \printbibliography, which accepts an optional argument. This argument can be used to filter the entries:

- by special fields (type or keyword),
- via the definition of categories in the preamble (using \DeclareBibliographyCategory), then assigning each entry to one of these categories (using \addtocategory),
- depending on the position of the citation inside the document, by using the refsection or refsegment options.⁵⁷

This allows easily dividing the bibliography by chapters or topics by using \printbibliography several times with different filters. When making multiple

⁵⁷ The sections can also be defined manually enclosing portions of the document between \begin{refsection} and \end{refsection}.

Table 3: List of the $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$ commands for the references in the text and their effect in the compact author-year style (option `author-year-comp`).

References in the text		
<code>\textcite{mori06}</code>	\Rightarrow	MORI et al. (2006)
<code>\textcite[chap.~4]{mori06}</code>	\Rightarrow	MORI et al. (2006, chap. 4)
References with parentheses		
<code>\autocite{mori06}</code>	\Rightarrow	(MORI et al. 2006)
<code>\parencite{mori06}</code>	\Rightarrow	(MORI et al. 2006)
<code>\parencite[chap.~4]{mori06}</code>	\Rightarrow	(MORI et al. 2006, chap. 4)
<code>\parencite[see] []{mori06}</code>	\Rightarrow	(see MORI et al. 2006)
<code>\parencite[see] [chap.~4]{mori06}</code>	\Rightarrow	(see MORI et al. 2006, chap. 4)
Multiple references		
<code>\cite{mori06,rossi07}</code>	\Rightarrow	MORI et al. 2008; ROSSI et al. 2007
<code>\textcite{mori06,rossi07}</code>	\Rightarrow	MORI et al. (2008); ROSSI et al. (2007)
<code>\parencite{mori06,rossi07}</code>	\Rightarrow	(MORI et al. 2008; ROSSI et al. 2007)
<code>\cite{mori06,mori08}</code>	\Rightarrow	MORI et al. 2006, 2008
<code>\cite{mori06a,mori06b}</code>	\Rightarrow	MORI et al. 2006a,b
References without parentheses		
<code>\cite{mori06}</code>	\Rightarrow	MORI et al. 2006
<code>\cite[chap.~4]{mori06}</code>	\Rightarrow	MORI et al. 2006, chap. 4

Table 4: List of the $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$ commands for the references in the text and their effect in the numbered style (option `numeric-comp`).

References in the text		
<code>\textcite{mori06}</code>	\Rightarrow	MORI et al. [6]
<code>\textcite[chap.~4]{mori06}</code>	\Rightarrow	MORI et al. [6, chap. 4]
References with parentheses		
<code>\cite{mori06}</code>	\Rightarrow	[6]
<code>\parencite{mori06}</code>	\Rightarrow	[6]
<code>\autocite{mori06}</code>	\Rightarrow	[6]
<code>\cite[chap.~4]{mori06}</code>	\Rightarrow	[6, chap. 4]
<code>\cite[see] []{mori06}</code>	\Rightarrow	see [6]
<code>\cite[see] [chap.~4]{mori06}</code>	\Rightarrow	see [6, chap. 4]
Multiple references		
<code>\cite{mori06,mori08}</code>	\Rightarrow	[3, 6]

bibliographies, a complete entry list can still be generated with the `\bibbysection`, `\bibbysegment` or `\bibbycategory` commands.

5.4 Commands for indexing

Automatic indexing of bibliographic entries is a very interesting feature of $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$. An index and an author index are very useful for any kind of book and thesis. $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$ can automatically add the authors cited in the text into an author index (or another type of index) by using the `indexing` option. $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$ uses external packages to create indexes: `makeidx` for basics operations and `index` for advanced features such as multiple indexes.

This section will show an example of automatic indexing; further details can be found in Lehmann (2009, in particular section 3.1.2 and the templates that come with the package documentation).

Assuming that our bibliographic database is called `database.bib`, the following code can be used to create an index with $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$:

```
\documentclass{...}

\usepackage[indexing]{biblatex}
\bibliography{database}

\usepackage{makeidx}
\makeindex

\begin{document}

As reported by \textcite{Kopka1995}...
\clearpage

As discussed by \citeauthor{Goossens1995} in...
\clearpage

\nocite{*}
\printbibliography
\printindex
\end{document}
```

If our document is called `document.tex`, we need to compile it in this order:

```
latex document
bibtex document
latex document
makeindex document
latex document
```

This produces a document with a bibliography followed by an index with the name of all the cited authors.

5.5 Multilingual bibliographies

BIB_{La}T_EX also includes some of `babelbib`'s features. When the `babel` option is used with one of the values `hyphen` or `other`, the package checks the `hyphenation` field for each bibliography entry. If a language is specified through this field, BIB_{La}T_EX uses the appropriate hyphenation rules and the translation for words such as “editor”, “volume”, etc. The translations used are stored in `.ltx` files that come with the package.

6 Acknowledgments

I would like to thank Massimiliano Dominici for writing section 5 about BIB_{La}T_EX and Gustavo Cevolani for writing section 5.4 about indexes with BIB_{La}T_EX. I also would like to thank Valeria Angeli, Claudio Beccari, Caterina Mori, and Gianluca Pignalberi for their suggestions during both the writing and the reviewing process of this article.

References

- Daly, P.W. “Customizing Bibliographic Style Files”, 2007a. <http://mirror.ctan.org/macros/latex/contrib/custom-bib/makebst.pdf>.
- Daly, P.W. “A Master Bibliographic Style File for numerical, author-year, multilingual applications”, 2007b. <http://mirror.ctan.org/macros/latex/contrib/custom-bib/merlin.pdf>.
- Daly, P.W. “Natural Sciences Citations and References (Author-Year and Numerical Schemes)”, 2009. <http://mirror.ctan.org/macros/latex/contrib/natbib>.
- Garcia, Federico. “L_AT_EX and the different bibliography styles”. *TUGboat* **28**(2), 2007. <http://tug.org/TUGboat/Articles/tb28-2/tb89garcia.pdf>.
- Kopka, H., and P. Daly. *A Guide to L_AT_EX — Document Preparation for Beginners and Advanced Users*. Addison-Wesley, fourth edition, 2003.
- Lehmann, Philipp. “The bibl_{La}TeX package”. 2009. <http://mirror.ctan.org/macros/latex/expt1/bibl_{La}TeX/doc/bibl_{La}TeX.pdf>.
- Patashnik, Oren. “BIB_{La}T_EXing”, 1998. <http://mirror.ctan.org/biblio/bibtex/contrib/doc/btxdoc.pdf>.
- UK TUG. *The UK TUG FAQ*, 2009. <http://www.tex.ac.uk/faq>.

◇ Lapo F. Mori
 Dipartimento di Ingegneria Meccanica,
 Nucleare e della Produzione
 Università di Pisa
 Pisa, Italy
 lapo dot mori (at) ing dot unipi dot it