Εὔτυπον, τεῦχος № 16–19 — Ὀκτώβριος/October 2007

Citations, reference list, and author index with apacite

Erik Meijer

University of Groningen, Faculty of Economics
and
RAND Corporation
E-Mail: apacite@gmail.com

The apacite package is a package that can be used with LaTeX and BibTeX to generate citations and a reference list, formatted according to the rules of the American Psychological Association. Furthermore, apacite contains an option to (almost) automatically generate an author index as well. A recent addition is the support of different languages. The package can be customized in many ways. This paper describes the apacite package, paying special attention to its idiosyncracies and how the problems associated with these have been solved.

1 Introduction

The American Psychological Association (APA) is very strict about the style in which manuscripts submitted to its journals are written and formatted. The requirements of the APA are described in the *Publication Manual of the American Psychological Association*, the latest version of which is the 5th edition (American Psychological Association, 2001). In the following, this is simply called the *APA manual*.

The APA manual discusses how candidate authors should write their manuscripts: writing style, parts of a manuscript and their order, presentation of the results in the form of tables and figures, and so forth. Candidate authors should study this and adhere to this.

The APA manual also gives specific rules about the formatting of a manuscript. This includes double spacing, a running head, the typographic style of section headings, the placement of tables and figures on separate pages at the end of the document, and so forth. LaTeX users will recognize these as "style" elements that should be defined in a package (`.sty` file) or class (`.cls` file). Their specific documents (`.tex` file) should be largely style-independent. This idea of separating content and logical structure from specific formatting is one of the basic elements of LaTeX (Lamport, 1994, p. 7).

An implementation of the formatting rules of the APA manual for use with LaTeX is the apa class (see the article by Athanassios Protopapas in this issue). This handles all kinds of issues about general document formatting, title page, section headings, figures and tables, and so forth.

An important part of the APA style is the way citations and the reference list should be formatted. This takes 75 pages in the APA manual (pp. 207–281, excluding the references to legal materials). This part is not handled by the apa class, but by the apacite package. This package consists of a LaTeX package (`apacite.sty`), a BibTeX style file (`apacite.bst`), extensive documentation, and several supplementary files. The apa class requires the apacite package, but apacite can be used without apa. This is especially convenient for manuscripts that should adhere to a style that is different from the APA style, but using a citation and reference style that is similar to the APA citation and reference style. In section 6, it is briefly discussed how the behavior of apacite can be adapted to (slightly) different styles.

The APA citation and reference style is an author-year citation style combined with an unnumbered, alphabetically ordered, reference list. Compared to other author-year styles, the APA style has a lot of different requirements, which are not always easy to implement in computer code. Examples of requirements that are specific to APA style are:

- If the first author of a cited work has the same surname as the first author of another cited work, but different initials, then the initials should also be given in the citation: "J. M. Goldberg and Neff (1961) and M. E. Goldberg and Wurtz (1972) studied ...".

- Like several other styles, the APA requires that when citing a work with three, four, or five authors, they should all be mentioned in the first citation in the text, but only the first followed by "et al." in subsequent citations. For six or more authors, the abbreviated ("et al.") form should be used in all citations. However, when there are citations to different works published in the same year, with the same first author but different subsequent authors, the ambiguity should be solved by adding authors to the citation until there is no ambiguity left: If the first citations are "Bradley, Ramirez, Soo, and Brown (1994)" and "Bradley, Soo, Ramirez, and Brown (1994)", then normally both would be abbreviated to "Bradley et al. (1994)" the second and later times they are cited. To distinguish them if they occur in the same document, they are abbreviated to "Bradley, Ramirez, et al. (1994)" and "Bradley, Soo, et al. (1994)", respectively.

- When the work used is a reprint or translation of an earlier work, this should be mentioned in the reference list, e.g.,

  Freud, S. (1961). The ego and the id. In J. Strachey (Ed. and Trans.), *The standard edition of the complete psychological works of Sigmund*

Freud (Vol. 19, pp. 3–66). London: Hogarth Press. (Original work published 1923)

In text, such a work must be cited using the form "Freud (1923/1961)".

- If the work referred to is a non-English-language source, the English translation of the title should be given in brackets after the title in the reference list.

The first version of apacite (1994) was based on the requirements of the third edition of the APA manual (American Psychological Association, 1984). The `apacite.sty` LaTeX package consisted for a large part of the citation part of `theapa.sty` by Young U. Ryu. It did quite a good job, but a couple of important aspects of the APA style were not formatted correctly. The current version has been changed and extended in many ways and therefore cannot be considered a straightforward adaptation of `theapa.sty` anymore, although it provides largely the same commands, such as `\citeA`, and peculiarities in the command definitions, such as the use of `<...>` for a prefix note and `[...]` for a postfix note (see section 4).

Although the `apacite.bst` BibTeX style started out as a slightly adapted version of `theapa.bst`, it has been redesigned and rewritten a couple of times and currently contains only a few small parts of the code of `theapa.bst`.

Since 2004, apacite is based on the requirements of the fifth edition of the APA manual and almost all problems with the APA style have been solved. There are, however, a few exceptions of rare and extreme examples. The apacite documentation includes all examples from the APA manual and thus shows how these can be formatted correctly. It shows how, by using advanced LaTeX and BibTeX "tricks", even the rare and extreme examples that are not formatted correctly by default can be formatted correctly. Recent versions of apacite also provide an easy way to generate an author index (almost) automatically, see section 8.

Given the generally satisfactory way in which apacite is able to follow the APA style, recent and future developments are primarily directed at offering additional features and solving compatibility problems with possibly conflicting packages. An important feature of recent versions (June 2005 and later) is that it offers support for non-English languages, although this is still limited. See section 7 for a description of the implementation of this feature.

Section 2 first gives an overview of how citation with LaTeX and BibTeX works in general, and with apacite in particular. Then, section 3 discusses in detail the most important issues that have been dealt with in the BibTeX bibliography style file `apacite.bst`. Section 4 focuses on the LaTeX part of the package, describing the citation commands defined in `apacite.sty`. Section 5 discusses specific issues about the contents of user-written bibliography database (`.bib`) files as necessary for apacite. Section 6 addresses customization, i.e., the adaptation of apacite to slightly different styles. In section 7, the language support feature is introduced. Section 8 shows how an author index can

be generated, using the slightly different `apacitex.bst` bibliography style, the LaTeX package `apacite.sty` with specific indexing options, and the *MakeIndex* program. Finally, section 9 addresses some compatibility issues and section 10 concludes with a preview of future developments in apacite.

2 Overview of citation with LaTeX, BibTeX, and apacite

This section gives a global overview of how citation with apacite works. In order to give some insight in this, it is convenient to first look at how citation works with standard LaTeX, without any citation packages loaded, and with BibTeX using the plain style.

Citing a work starts with putting the information describing the work in a BibTeX database file, which is a simple (ASCII) text file with extension `.bib`. Here is a short file (`db1.bib`) with only one work (or *entry*):

```
@article{MaBaMc1988,
  author  = {Marsh, Herbert W. and Balla, John R.
             and McDonald, Roderick P.},
  year    = {1988},
  title   = {Goodness-of-Fit Indexes in Confirmatory Factor Analysis:
             The Effect of Sample Size},
  journal = {Psychological Bulletin},
  volume  = {103},
  pages   = {391--410},
}
```

The meaning is straightforward: This entry describes a journal *article*. The information about this entry is stored in several *fields*. The information of each field is given in the form "⟨*field*⟩ `= {`⟨*value*⟩`},`". When there are multiple authors (or editors), they are separated by the word "`and`" and surrounding space. The string "`MaBaMc1988`" is the *citation key*, which is a unique identifier that is used to link the citation in the LaTeX document to the entry in the BibTeX database file. Section 5 below discusses the content of the `.bib` file more extensively.

Here is a short LaTeX document (`doc1.tex`) that cites this entry, using only standard LaTeX and requesting the BibTeX style plain:

```
\documentclass{article}
\begin{document}
Marsh et al.\ \cite{MaBaMc1988} found that fit indexes may be unreliable.
But some are better than others \cite{MaBaMc1988}.
\bibliographystyle{plain}
\bibliography{db1}
\end{document}
```

The `\cite` command has two functions: It notifies BibTeX that the work "`MaBaMc1988`" is cited and thus must be included in the reference list, and (with standard LaTeX) it inserts a citation string of the form "[⟨*number*⟩]" in

the text, where ⟨*number*⟩ is the sequence number of the work in the reference list. In this example, only one work is cited in the document, which thus has sequence number 1, and so the `\cite` command inserts "[1]" in the text.

The `\bibliographystyle` command requests the BibTeX style plain, implemented in the file `plain.bst`. The `\bibliography` command tells BibTeX that it should search the database file `db1.bib` for entries corresponding to the citations, and it includes the formatted reference list at this point in the text.

However, in the first LaTeX run, the reference list has not been generated yet. Consequently, it cannot be included in the text yet, and the sequence number of the work is also not yet known. This is diagnosed by the following lines in the log file (`doc1.log`):

```
LaTeX Warning: Citation 'MaBaMc1988' on page 1 undefined on input line 4.
No file doc1.bbl.
```

The `.dvi` file (or `.pdf` file if pdfLaTeX is used) now does not have a reference list, and the intended citations are replaced by question marks, because of the missing information.

The communication from LaTeX to BibTeX is channeled through the `.aux` file. BibTeX does not search the `.tex` file for clues, but only reads the `.aux` file. After the first LaTeX run on the document above, the `.aux` file (`doc1.aux`) contains the following lines:

```
\citation{MaBaMc1988}
\citation{MaBaMc1988}
\bibstyle{plain}
\bibdata{db1}
```

As may be inferred from this, the `\cite` command writes the `\citation` lines, the `\bibliographystyle` command writes the `\bibstyle` line, and the `\bibliography` command writes the `\bibdata` line. Although `\citation`, `\bibstyle`, and `\bibdata` look like LaTeX commands (and in fact are LaTeX commands), they have no meaning to LaTeX. They are only used by BibTeX to extract the necessary information about bibliography style file, bibliography database file(s), and which works from the database(s) are cited.

Hence, after this first LaTeX run, BibTeX must be invoked to handle this information and produce the reference list. The BibTeX run corresponding with the current example generates the file `doc1.bbl`, with the following contents:

```
\begin{thebibliography}{1}

\bibitem{MaBaMc1988}
Herbert~W. Marsh, John~R. Balla, and Roderick~P. McDonald.
\newblock Goodness-of-fit indexes in confirmatory factor analysis: The effect
  of sample size.
\newblock {\em Psychological Bulletin}, 103:391--410, 1988.

\end{thebibliography}
```

Although it has the `.bbl` extension, it is a LaTeX file. We see that BibTeX has generated a thebibliography environment, which is a special kind of enumerate environment. Hence, this is a numbered list, and in standard LaTeX the numbers are put between square brackets: "[1]". The (mandatory) argument of the environment contains a number that is used to determine the width of the widest label. Here, the "`1`" means that "[1]" is the widest label.

The bibliography uses `\bibitem` instead of `\item` to start a new list item. The `\bibitem` has one mandatory argument, which is the citation key. This is used later to pass information to LaTeX such that the sequence numbers can be inserted by the `\cite` command. The remaining text is simply the formatted reference list entry.

Now that the reference list has been generated, LaTeX must be run a second time. This time, the `\bibliography` command reads the file `doc1.bbl`. Consequently, the resulting `.dvi` file now contains the reference list. However, because the reference list is typically inserted at the end of the document, and because of how LaTeX processes it, the sequence numbers are not yet available to the `\cite` commands. This implies that after the second LaTeX run, the LaTeX warning about the undefined citation still appears, but the message "`No file doc1.bbl.`" is now replaced by "`(./doc1.bbl)`", indicating that the reference list is included.

Furthermore, `doc1.aux` now contains the following additional line:

```
\bibcite{MaBaMc1988}{1}
```

This line has been written by the `\bibitem` command. It tells LaTeX that the sequence number corresponding to the work "`MaBaMc1988`" is 1. This also explains why the `\bibitem` command needs the mandatory argument.

Now, LaTeX must be run a third time. In the third run, the `.aux` file generated by the second run is read by TeX at `\begin{document}`. Hence, the `\bibcite` command in the `.aux` file is executed. This command stores the value "1" in the internal command "`\b@MaBaMc1988`". The `\cite` commands in the document insert "`[\b@MaBaMc1988]`" in the text, which now reduces to the desired "[1]".

After the third LaTeX run, the LaTeX warning has disappeared from the `.log` file as well and the `.aux` file is identical to the previous one. The resulting typeset document is shown in Figure 1. In summary, the following sequence of programs must be run: (1) LaTeX, (2) BibTeX, (3) LaTeX, (4) LaTeX. The result is that the `\cite` command puts "[1]" in the text and that BibTeX generates a numbered formatted reference list with labels of the form "[1]".

In order to comply with the rules of the APA manual, there are several things that apacite must do different than the standard LaTeX and plain implementation discussed thus far:

1. Numbered citations should be replaced by author-year citations. There are two forms: parenthetical citations of the form "(Author, year)" and in-text citations of the form "Author (year)". This means that the author

Marsh et al. [1] found that fit indexes may be unreliable. But some are better than others [1].

References

[1] Herbert W. Marsh, John R. Balla, and Roderick P. McDonald. Goodness-of-fit indexes in confirmatory factor analysis: The effect of sample size. *Psychological Bulletin*, 103:391–410, 1988.

Figure 1: Typeset document using bibliography style `plain`.

and year information about the cited work must be known to LaTeX at the point of citation. Furthermore, the `\cite` command must be redefined to produce the different formatting, and there must be two forms of `\cite` commands to enable the two citation forms.

2. The formatting of the author list in a citation with 3–5 authors depends on whether the current citation is the first citation to this work in the document or a subsequent citation. In the former case, all authors must be listed, whereas in the latter case, only the first author must be given, followed by "et al.". This means that the author information about the cited work that must be known to LaTeX at the point of citation consists of a "full" and a "short" author list, and that LaTeX must keep track of whether the current work has already been cited.

3. The reference list must be an unnumbered list, without item labels, but with a hanging indent.

4. The entries in the reference list must be formatted differently in several ways: initials instead of full given names, initials must be put *after* the surnames, the year comes immediately after the authors and between parentheses, and the volume of the journal must be italic and followed by a comma instead of a colon. There are numerous other differences that do not show up in this example.

These points more or less hold for all author-year citation styles. Hence, the problems that are faced by, and solutions that are implemented in, apacite are similar to those of other author-year styles, like natbib, harvard, or chicago.

The last point is fairly straightforward. The APA rules are very detailed and require much information to be inserted in the reference list, and the same information must be formatted differently in (only slightly) different situations, so it requires more code than in `plain.bst`, but it does not need any intricate interplay between LaTeX and BibTeX or complicated code.

Point 3 is also fairly standard. It requires a couple of redefinitions of the

thebibliography environment and the `\biblabel` command, but nothing complicated.

The first two points pose more problems. These have been largely solved in predecessors of apacite, most notably theapa. Therefore, in apacite, this code has been adapted, frequently considerably so, and extended in many ways, but the basic ideas and concepts underlying the solutions are still the same.

The two forms of citation are handled by providing the `\citeA` command for in-text citations in addition to `\cite`, which is used for parenthetical citations. In addition to these, apacite offers several other citation commands that are useful for specific purposes. These are described in section 4. Because of the provision of the `\citeA` command and the necessary (re)definitions of several LaTeX commands, apacite consists of a LaTeX package (`apacite.sty`) in addition to the BibTeX style file (`apacite.bst`). Thus, the LaTeX document in the example above is changed accordingly. The resulting file (`doc2.tex`) is:

```
\documentclass{article}
\usepackage{apacite}
\begin{document}
\citeA{MaBaMc1988} found that fit indexes may be unreliable.
But some are better than others \cite{MaBaMc1988}.
\bibliographystyle{apacite}
\bibliography{db1}
\end{document}
```

The `\usepackage` requests the apacite LaTeX package, whereas `\bibliographystyle` requests the apacite BibTeX style. Note that if the apa document class is used instead of article, the apacite LaTeX package is automatically loaded, the bibliography style is automatically set to `apacite`, and the `\bibliographystyle` command is disabled for user purposes. Furthermore, note that in this example document, the authors are not literally typed out anymore for the in-text citation. The `\citeA` command handles this. This is more in the LaTeX spirit of leaving explicit formatting that is style-dependent to specific packages or classes. Finally, note that the same bibliography database file is used as in the first example.

The resulting reference list (`doc2.bbl`) output by BibTeX is now

```
\begin{thebibliography}{}

\bibitem[\protect\citeauthoryear{Marsh, Balla, \BBA{} McDonald}{%
         Marsh et al.}{1988}]{MaBaMc1988}
Marsh, H.~W., Balla, J.~R., \& McDonald, R.~P.
\newblock (1988).
\newblock Goodness-of-fit indexes in confirmatory factor analysis: The
  effect of sample size.
\newblock \emph{Psychological Bulletin}, \emph{103}, 391--410.

\end{thebibliography}
```

In fact, the actual `.bbl` file looks quite differently. It contains many more (and different) line breaks, and several command strings (e.g., `\BCBL`, `\BOthers`, and

`\APACyear`). The former are inserted by `apacite` because BibTeX sometimes puts a line break in an undesired position (e.g., in the middle of a command name), whereas the latter have technical reasons and/or allow customization by the user. In order to keep the example readable, here and in several places below, generated files have been manually edited in a way such that the parts that are important for the discussion have been kept. The `\protect` and `\BBA` commands will be discussed below.

Apart from the different formatting of the reference list entry, the most important characteristic of this `.bbl` file is that the `\bibitem` command uses an optional argument. The contents of this argument are a `\citeauthoryear` command that itself has three arguments: the full author list, the short author list, and the year. This optional argument takes the place of the sequence number used with `plain.bst`. At this point, it does not output any text, so that there is no label in the reference list. However, this command with its arguments is copied to the `.aux` file in subsequent LaTeX runs. Here is the corresponding excerpt from `doc2.aux` after the second LaTeX run:

```
\bibcite{MaBaMc1988}{\citeauthoryear{Marsh, Balla,
                    \BBA{} McDonald}{Marsh et al.}{1988}}
```

The "1" from the first example is now replaced by the `\citeauthoryear` command and its arguments. The citation commands are able to extract the arguments from the `\citeauthoryear` command, so that the full or short author list and/or the year can be inserted in the output with the relevant formatting. This is achieved because `\b@MaBaMc1988` is now defined to be `\citeauthoryear{Marsh, Balla, \BBA{} McDonald}{Marsh et al.}{1988}` and the citation commands redefine `\citeauthoryear` in a suitable way before executing `\b@MaBaMc1988`. Clearly, it is important that the `\citeauthoryear` command must not be executed before it needs to be used. The `\protect` command in the `.bbl` file ensures this.

The `\BBA` command becomes "and" in in-text citations and "&" in parenthetical citations, as required by the APA manual. Whether the full or short author list must be used depends on whether the current citation is the first to the current work in this document. This situation is diagnosed by the command `\flag@MaBaMc1988`. Initially, this command is not defined. The citation commands check whether it is defined and then define it. Hence, if this work was cited before, `\flag@MaBaMc1988` has already been defined at the beginning of the citation command. Thus, by checking whether this command has been defined, the citation commands actually check whether this work has been cited before.

The resulting typeset document is shown in Figure 2. This shows that indeed the full author list is used in the first citation, whereas the short one is used in the second citation.

Marsh, Balla, and McDonald (1988) found that fit indexes may be unreliable. But some are better than others (Marsh et al., 1988).

References

Marsh, H. W., Balla, J. R., & McDonald, R. P. (1988). Goodness-of-fit indexes in confirmatory factor analysis: The effect of sample size. *Psychological Bulletin*, *103*, 391–410.

Figure 2: Typeset document using bibliography style `apacite`.

3 The bibliography style file `apacite.bst`

As stated above, the reference list is generated by BibTeX, using a certain bibliography style implemented in a `.bst` file, which must be explicitly requested by the user (unless it is explicitly requested by a package or class file, such as `apa.cls`). When BibTeX is invoked, it reads the `.aux` file to check which style file must be used, which `.bib` files must be searched for referenced works, and which works must be included in the reference list. Then BibTeX executes the program that the `.bst` file contains. This program performs the following tasks: (1) read the cited entries from the `.bib` file; (2) sort the entries; (3) generate citation labels (the optional argument of `\bibitem`; for `apacite`, this primarily means determining the arguments of the `\citeauthoryear` command); (4) format the reference list entry.

Task (1) consists of the single `READ` command. It produces a linear list of records, where each record corresponds with a cited work. Each record has a number of fields, which are the fields from the `.bib` file. The list is in order of first citation in the document. Tasks (2) and (3) are discussed in some detail in section 3.1 below and Task (4) is discussed in section 3.2.

Consequently, although the `.bst` file is usually called a bibliography style file, suggesting that it only contains a list of options and/or parameters, it actually contains a computer program implementing a style. The program is written in BibTeX's programming language, which is typically called the "bst language". This language looks quite different from programming languages like C or Pascal and programs in it are therefore hard to read for the uninitiated. In technical terms, it is a reverse Polish stack-based language. In practice, this means that statements must often be read from right to left (or bottom to top) and that arguments of functions are not explicitly stated, but assumed to be on the stack when the function is called. Examples are shown in Mittelbach and Goossens (2004, section 13.6).

The restrictions of BibTeX in general and the bst language in particular have caused some technical difficulties in the development of `apacite`. Some of the problems are caused by the fact that BibTeX does not allow dynamic

memory allocation and therefore, the maximum number of variables etc. have to be chosen at compile time. For example, the original BibTeX source code allowed only 10 global string variables. The current version of `apacite.bst` uses 20, which appeared to be the maximum number allowed by the BibTeX in my teTeX distribution under Cygwin, but apparently there still exist TeX distributions with more limited BibTeX versions. Such distributions are also likely to have problems with other aspects of the size of `apacite.bst`, e.g., the number of functions it defines. Other problems with the bst language are that it does not have arrays and that the elements of the reference list cannot be randomly accessed. This means that operations on the elements can only be done in a complete forward pass through the list, through the `ITERATE` command, or a complete backward pass through the list, through the `REVERSE` command. The only other operation on the list is sorting, through the `SORT` command, using the single sorting key `sort.key$`. Furthermore, the `ITERATE`, `REVERSE`, and `SORT` commands are only allowed at the top level, so that something like

```
WHILE (reference list is not yet correct)
  ITERATE (improvement function)
```

is not allowed. To circumvent the restrictions imposed by BibTeX and at the same time improve the compliance with the complicated rules of the APA manual, `apacite.bst` has been redesigned and largely rewritten a couple of times. The remainder of this section describes how some of the APA-specific issues have been solved.

3.1 Generating the citation labels and sorting the reference list

About half of `apacite.bst` is devoted to generating the citation labels and sorting the reference list. This is much more than with other BibTeX styles and is due to the complicated rules of the APA manual and the technical restrictions imposed by BibTeX. The APA manual requires that the reference list be sorted by author, year, and title. Within author, the first author is sorted first, then the second, and the initials of the first author take precedence over the surname of the second author, and so forth. As usual, shorter names are sorted before longer. For example, a reference in which the surname of the first author is "Brown" comes before a reference in which it is "Browning", regardless of their initials, regardless of whether there is a second author or not, and regardless of the name of the second author.

BibTeX is not able to sort on multiple sort keys, unlike a typical spreadsheet program, in which rows can be sorted by using multiple columns to determine the sorting order. Fortunately, however, because BibTeX's sorting routine does not compress multiple spaces into one space (like TeX does), and a space comes before anything else, it is fairly straightforward in principle to implement the precedence rules by varying the number of spaces between the different parts. Thus, `apacite.bst` puts two spaces between a surname and the corresponding

initials (which are put after it), three spaces between authors, and four spaces between author(s) and year and between year and title. This ensures that the surname of the first author takes precedence over anything else, the initials of the first author are used before the surname of the second author ("J. S. Brown" comes after "J. Brown & D. Jones"), and so forth.

This implies, however, that the sorting key can become very long. Including the whole title in particular tends to increase the lengths of some sorting keys considerably. In first implementation attempts, this frequently led to errors because of exceeding BibTeX's global maximum string size. This is a fixed number that is chosen in the BibTeX source code, `bibtex.web`. BibTeX executables included in more recent TeX distributions have set this and other BibTeX restrictions at higher levels than in the original source code and (thus) older versions of BibTeX. Nevertheless, to reduce the probability of such problems, `apacite.bst` first sorts the reference list on the title field alone and stores the resulting sequence number of each reference. Then, in sorting, each title is replaced by this number, converted to a string, with leading zeros if necessary. If the title starts with one of the articles "A", "An", or "The", these are ignored in sorting the titles. It would be desirable to use a different list for non-English languages, but such an advanced language option is not yet available. If the title field is missing, several other fields are checked instead.

Next, the citation labels are determined. These are the strings that are included in the text by the citation commands, i.e., the arguments of the `\citeauthoryear` command. The first argument is the full author list for the first citation. Typically, this is of the form

Smith	(1 author)
Smith and Jones	(2 authors)
Smith, Jones, and Baker	(3 authors; 4 and 5 authors are formatted analogously)
Smith et al.	(more than 5 authors).

The second argument is the short author list for subsequent citations. This is the same as the full author list, except when there are 3–5 authors, in which case this argument has the "et al." form. The third argument is typically just the year of publication.

There are a couple of exceptions, however. If there is a reference for which J. Smith is the first author and another reference for which P. Smith is the first author, then "Smith" in the above must be replaced by "J. Smith" or "P. Smith", respectively. This is regardless of any co-authors and regardless of whether the years are the same or not.

Another exception is when there are multiple reference list entries with the same author list, published in the same year. Then, in the third argument, a suffix is added to the year, so that we get, e.g., "2002a" and "2002b".

The third exception is the most complicated one: If two author lists are different, but both reduce to the same short author list, and are published in the same year, then there is an ambiguity that must be resolved by adding authors

to the short (and possibly full) author list until the ambiguity disappears. So "Smith, Jones, and Baker (2002)" and "Smith, Baker, and Jones (2002)" are not abbreviated in the short list. With "Smith, Jones, Baker, Taylor, and White (2002)" and "Smith, Baker, Jones, and Brown (2002)", the short forms become "Smith, Jones, et al. (2002)" and "Smith, Baker, et al. (2002)", respectively.

In order to handle these situations correctly, an intermediate sorting is done in `apacite.bst`. This sorting is done with the following precedence orders: (1) last name of first author, (2) initials of first author, (3) last name of second author, or "zzzz" (representing "et al.") if there are more than two authors, (4) year, (5) last names of second and subsequent authors if there are more than two authors, (6) title number. This sorting order ensures that any ambiguous cases that would ordinarily lead to the same short citation immediately follow each other. This is not the case if the final sorting order would be used. For example, "Smith and Brown (2003)" comes between "Smith, Baker, Jones, and Brown (2002)" and "Smith, Jones, Baker, Taylor, and White (2002)" in the final reference list. Keeping ambiguities together is important, because BibTeX does not allow random access to elements of the list of references, only forward or backward passes through the whole list. Therefore, and because BibTeX does not have arrays, comparisons between different references can only be made by storing information from the previous entry in (a few) global strings and comparing information from the current entry with these strings.

Given this intermediate sorting, it follows that if an entry has the same author list and year as the previous one, then a year suffix must be added. If the previous entry did not (yet) have a year suffix, the current one is "b". Otherwise, the current one takes the letter from the alphabet following the one for the previous suffix. By a subsequent backward pass through the list, the suffix "a" is added when necessary. Actually, `apacite.bst` uses a number instead of a letter, and puts it in the argument of a command. This allows different suffix forms defined by the user. But "a", "b", etc. is the default, required by the APA manual.

Similarly, if the first author of the current entry has the same surname as the first author of the previous entry, but different initials, then the initials must be added. Again, a backward pass fixes the first such author.

Finally, if the short citation form of the current entry would be the same as the short citation form of the previous entry, but the author lists are different, then authors are added until the ambiguities are resolved. The backward pass is a little more involved now than with the previous two situations, because it can also change entries that had already been changed in the forward pass. Consider, for example, the following three entries: (1) Smith, Baker, Jones, and Brown (2002), (2) Smith, Jones, Baker, Taylor, and White (2002), (3) Smith, Jones, Taylor, Baker, and White (2002). Then in the forward pass, the first one gets the tentative label "Smith et al. (2002)", then the second becomes "Smith, Jones, et al. (2002)", and the third becomes "Smith, Jones, Taylor, et al. (2002)". In the backward pass, the second is changed into "Smith, Jones,

Baker, et al. (2002)" and the first one is changed into "Smith, Baker, et al. (2002)".

Now, the final citation labels have been generated and the reference list is sorted in its final order. Thus, `apacite.bst` sorts the reference list three times, whereas most other BibTeX styles only sort it once or twice (or not at all).

A couple of final remarks must be made here. The discussion above assumed that the cited work actually has an author (or editor acting as author). This is not always the case. If there is no author, the APA manual requires that the title must be put in the author position in the reference list and that (an abbreviation of) the title must be used instead of an author in the citation in the text. Thus, if there are no authors or editors acting as authors, apacite uses the title. If there is no title as well, a couple of other fields are checked.

Whether or not an author is available, the user can override sorting and citation by using the key and/or firstkey fields. When both of these are available, the contents of the firstkey field are used as full author list and the contents of the key field are used as short author list for citing purposes. The firstkey then also determines the sorting of the entry. When one of these two fields is present but the other not, the available one is also used in the place of the unavailable one. The apacite documentation contains some examples of situations in which the key and/or firstkey fields can be exploited fruitfully to format citations and sort entries correctly.

BibTeX typically handles accents and other nonstandard characters from Western-European languages well, at least in English documents: "`\"{u}`" is treated as "`u`", "`\ss`" is treated as "`ss`", and "`\ae`" is treated as "`ae`". However, in non-English languages this sorting may not be appropriate and one may, for example, desire that "`\"{u}`" is treated as "`ue`". Numerous problems are caused by different conventions in different languages. Some examples are given in the documentation of the amsrefs package. Non-latin scripts (Greek, Hebrew, Arabic, Chinese, etc.) may not be handled correctly as well. I am not able to judge this, nor does it seem to be a problem that must be solved in apacite. It has been announced that BibTeX 1.0, when it will be released, will support non-English languages as well.

If one only has a few sorting problems, then BibTeX can be tricked by using the `\APACSortNoop` command. This is a LaTeX command provided by apacite. It has one mandatory argument, but it outputs nothing. However, BibTeX does not know this and uses the contents of this command to sort the entry. For example,

```
author = {M{\"{u}}ller, Gerd},
```

is sorted as "`muller␣␣g`", whereas

```
author  = {{\APACSortNoop{Mueller}}M{\"{u}}ller, Gerd},
```

is sorted as "`muellermuller␣␣g`", which typically results in the correct ordering. In the typeset document, both look like "Müller". Note that the extra

pair of braces around the `\APACSortNoop` command are necessary to suggest to BibTeX that it is an "accent". The specification

```
author  = {\APACSortNoop{Mueller}M{\"{u}}ller, Gerd},
```

is sorted as "`apacsortnoopmuellermuller␣␣g`", which is incorrect. Of course, when one writes a document in Greek, referring to many authors using the Greek script, using `\APACSortNoop` to achieve the correct sorting would not be very convenient. Apparently, when using the ISO-8859-7 encoding, both the Greek and the Latin entries are sorted correctly, but all Latin entries precede all Greek entries. This may or may not be acceptable.

3.2 Reference list formatting

Most of the formatting of the reference list is fairly straightforward. There are many differences with most other bibliography styles, but most of these are small and straightforward formatting differences, like "Address: Publisher" instead of "Publisher, Address". However, because the APA manual has a large number of peculiarities (e.g., it requires "Address: Publisher" for a book, but "Organization, Address" for unpublished manuscripts), the code is much longer than for other styles.

Some of the idiosyncratic requirements of the APA manual have led to the introduction of fields in the `.bib` file that are supported by apacite. See section 5 for a description of these fields. Evidently, the support of these fields requires code in `apacite.bst` that formats these fields. But this code is fairly simple as well.

Another complication of the APA requirements is that the title (name) of a computer program or software and/or its manual, a programming language, and a message to a forum or a newsgroup, should not be formatted in italics. These kinds of references would normally be put in a `@misc` entry, although manuals would also frequently be put in a `@manual` entry, but this reduces to `@misc` in apacite. However, titles of other kinds of references that are also typically put in a `@misc` entry, such as internet pages and motion pictures, must be formatted in italics. Rather than defining different entry types for these, which would make apacite less compatible with other bibliography styles, this is solved in apacite by using the type field. If the contents of this field are `\bibmessage`, `\bibcomputersoftware`, or one of a couple of related commands, `apacite.bst` formats the title in upright instead of italics. Adding a type field to a `@misc` (or `@manual`) entry in a `.bib` file will probably do not any harm with other bibliography styles, so this is probably only a minor nuisance.

An apacite feature that shows up in the `.bbl` files, but has been edited out of the stylized `.bbl` files for the examples in section 2, is that `apacite.bst` inserts an `\APACinsertmetastar` command at the beginning of each reference list entry, i.e., directly after (the arguments of) the `\bibitem` command, before the authors of the entry in the reference list. For the entry in the examples, this would look like

```
\APACinsertmetastar{MaBaMc1988}
```

In a manuscript describing a meta-analysis, i.e., a quantitative analysis using results from other studies as data, the studies that have been used in the meta-analysis must be included in the reference list, preceded by an asterisk (⋆). The `\APACinsertmetastar` inserts this asterisk if the entry `MaBaMc1988` was used in the meta-analysis, and omits it otherwise. In section 4 below, it is explained how this is done.

The APA manual requires that in the reference list, titles of works are formatted in sentence-style. That is, the first word of the title (and of the subtitle, when present) starts with a capital, but all other words are in lower case, except for proper names, acronyms, etc. Because many other styles require that all important words of titles (or sometimes only of titles of standalone works such as books) are capitalized, it is best to capitalize the words in the `.bib` file. `apacite.bst` uses a BibTeX command (`change.case$`) that converts strings to sentence-style. Of course, BibTeX does not know which words are proper names. Therefore, the user has to "protect" such words (or their first letters) by including them between an additional pair of braces. Thus,

```
title = {Cognitive Functions of Centenarians: The {Tokyo}
         {Metropolitan} {Centenarian} {Study}},
```

is formatted as *Cognitive functions of centenarians: The Tokyo Metropolitan Centenarian Study*, which is desired, whereas

```
title = {Cognitive Functions of Centenarians: The Tokyo
         Metropolitan Centenarian Study},
```

is formatted as *Cognitive functions of centenarians: The tokyo metropolitan centenarian study*, which is incorrect. Note that BibTeX knows that after a colon, a subtitle is started, so that the 'T' of "The" is always capitalized. Unfortunately, however, this is not the case with the question mark ('?') and the exclamation point ('!'). So it must be remembered that if the main title ends with a question mark or an exclamation point, the first word of the subtitle must be protected as well. The same holds when the main title ends with a period ('.'), but this makes more sense, because a period is used for many other purposes than ending a title.

BibTeX does not support options. Therefore, it is not straightforward to change the capitalization behavior of apacite (as users often request) without conflicting with APA style. However, the implementation described above is especially inconvenient for German users. In German, all nouns are capitalized. Therefore, users would have to protect each noun of each German title. Partly because of the introduction of the language support in apacite (see section 7), this has become a pressing problem. In experimental versions of apacite, this has been implemented since 2006 by inserting

```
\APACrefbtitle{Cognitive Functions of Centenarians}{%
               Cognitive functions of centenarians}
```

in the `.bbl` file (or `\APACrefatitle` for articles). That is, the title is inserted twice: once capitalized and once in sentence style. The default still gives the sentence-style title, but by redefining the command, e.g.,

```
\renewcommand{\APACrefbtitle}[2]{#1}
```

the capitalized version is obtained. Moreover, this redefinition can in principle be done locally, i.e., separately for each entry. Thus, German titles could be capitalized, whereas English titles are in sentence style. However, it will require some tricks by the user to implement such sophisticated behavior. Future versions of apacite may provide easier support for this.

4 The citation commands

As mentioned above, standard LaTeX only provides the `\cite` command for citation, whereas author-year styles need at least two. Therefore, packages like natbib, harvard, chicago, amsrefs, and apacite define additional citation commands. Unfortunately, they use different commands, so that a LaTeX document that is intended to be used with one of these packages cannot be used with another by just changing the `\usepackage` line. The commands that are defined in `apacite.sty` are the commands that are also used in `theapa.sty`, and straightforward extensions thereof. This is for historical reasons, because apacite started out as an adaptation of theapa and because theapa also aimed at implementing the rules of the APA manual. The apacite documentation has stated for some time that "future versions of apacite will support the natbib citation commands", but this still has not been implemented yet.

The basic citation commands provided by apacite are `\cite`, `\citeA`, `\citeauthor`, and `\citeyear`. Analogous to the standard LaTeX `\cite` command, these commands have one mandatory argument, which is a list of keys that identify which works are cited. The commands have two optional arguments. The first optional argument, included between '`<`' and '`>`' signs, is a prefix, which is to be included before the actual citations. The second optional argument, included between '`[`' and '`]`' signs, is a postfix, which is to be included after the actual citations. `\cite` gives a parenthetical citation, `\citeA` provides an in-text citation, `\citeauthor` only cites the authors, and `\citeyear` only cites the year (within parentheses). Additionally, apacite defines the `\citeNP` and `\citeyearNP` commands. Here, `NP` stands for "no parentheses". They are equivalent to the versions without `NP`, except that the parentheses are left out.

The usage and results of these citation commands are illustrated in Table 1. It is clear that not all possibilities are useful for each command, but they are provided to make the structure of the commands the same, which is convenient for both the user and for the design of the commands.

In addition to the citation commands mentioned, there are also `full` and `short` versions of them (except `\citeyear` and `\citeyearNP`), e.g., `\fullcite` and `\shortcite`. When a cited work has 3–5 authors, the basic citation commands give the full list of authors the first time a work is cited, but only the

Table 1: Examples of usage of basic citation commands

Command	Result
`\cite<e.g.,>[p.~11]{Jone01,Ross87}`	(e.g., Jones, 2001; Ross, 1987, p. 11)
`\citeA<e.g.,>[p.~11]{Jone01,Ross87}`	e.g., Jones (2001); Ross (1987, p. 11)
`\citeauthor<e.g.,>[p.~11]{Jone01,Ross87}`	e.g., Jones; Ross, p. 11
`\citeyear<e.g.,>[p.~11]{Jone01,Ross87}`	(e.g., 2001; 1987, p. 11)
`\citeNP<e.g.,>[p.~11]{Jone01,Ross87}`	e.g., Jones, 2001; Ross, 1987, p. 11
`\citeyearNP<e.g.,>[p.~11]{Jone01,Ross87}`	e.g., 2001; 1987, p. 11

first author and "et al." in subsequent citations. The `full` versions always give the full list in this case, whereas the `short` versions always give the abbreviated list. For one or two authors, all citation commands always give all authors, whereas for six or more authors, all citation commands always give the abbreviated form. More details about the full and short author lists have been given in section 3.1.

Furthermore, apacite supports the general LaTeX `\nocite` command. However, the APA manual requires that only works that are cited in the text should be included in the reference list, which implies that `\nocite` should not be used in manuscripts submitted to APA journals. There is one exception to this rule. As mentioned in section 3.2, in a manuscript describing a meta-analysis, the studies that have been used in the meta-analysis do not have to be cited, but they must be included in the reference list, preceded by an asterisk. Instead of tweaking the `\nocite` command itself, apacite supplies the `\nocitemeta` command for this purpose. In addition to the `\nocite` command, `\nocitemeta` defines a command `\flagmeta@`⟨*cite-key*⟩, which is checked by the `\APACinsertmetastar` command already mentioned in section 3.2. In this way, it is determined whether the asterisk must be inserted in the reference list or not. The `\nocitemeta` command also redefines an internal command `\APAC@metaprenote`, so that when `\nocitemeta` is used in the document, a message appears at the beginning of the reference list. This message explains the meaning of the asterisks, as required by the APA manual.

5 Contents of the `.bib` file

The information that is used by LaTeX/BibTeX to generate the citations and reference list must be stored by the user in one or more files with the `.bib` extension. A detailed overview of the contents of the `.bib` file is given in Kopka and Daly (2004, section 12.2), Mittelbach and Goossens (2004, section 13.2), and in the apacite documentation. The six most frequently used entry types and their typical fields are listed in Table 2.

Note that a book typically has an author *or* an editor, whereas a chapter in a

Table 2: The typical fields of six frequently used entry types.

article	book	incollection	techreport	phdthesis	misc
author	author	author	author	author	author
	editor	editor			
year	year	year	year	year	year
title	title	title	title	title	title
			type	type	type
journal		booktitle			
volume					
number			number		
pages		pages			
	address	address	address	address	address
	publisher	publisher	institution	school	publisher

book (`@incollection`) typically has both. The title field of an `@incollection` gives the title of the article (chapter) in the book, whereas booktitle gives the title of the whole book. Historically, in BibTeX, a `@book` has a publisher, whereas a `@techreport` is published by an institution, and this role is played by a school for theses and by an organization for an `@unpublished`.

To be able to obey the rules of the APA manual, apacite provides several fields and reference types that are not described in the standard BibTeX documentation. Furthermore, the meaning and usage of many fields and reference types that *are* described there have been altered somewhat.

The following reference types are defined in apacite but not available in other BibTeX styles: `@magazine` (for a magazine article), `@newspaper` (for a newspaper article), `@intechreport` (for an article in an edited report), and `@lecture` (for a seminar, a paper presented at a meeting or a conference, etc.). Finally, apacite provides the `@literal` reference type. If the other categories do not format the item correctly, this category can be used. The idea is that the user supplies the complete formatting of the reference. This is, however, rarely necessary, because most uncommon references can be formatted correctly with the `@misc` reference type. The `@literal` has been mainly introduced to be able to format references to legal court cases correctly. The APA manual discusses these, but they have not been explicitly implemented in apacite. The camel suite provides extensive support for citing court cases, so authors who frequently cite court cases may want to have a look at this suite.

The following fields are defined in apacite but are not available in (most) other BibTeX styles:

chair The chair(s) of a symposium or meeting. Used for lectures.

day The day of the month on which the referenced item was published, produced, or presented. Mainly used for articles in daily or weekly magazines

or newspapers, for lectures (although the APA manual only specifies the month), and electronic documents.

englishtitle The English translation of the title of an item with a non-English title. The APA manual requires that if the referenced work has a non-English title, an English translation should be given as well. Because apacite now contains some support for other languages of the main document (see section 7), the meaning of this field will probably change somewhat in a future version of apacite.

firstkey The firstkey field, if not empty, is used as "author" for the first citation to an item. Subsequent citations then use the key field. This can be used if there is no author or editor field that can be used for citations, or in certain cases with corporate authors, where the citation in the text uses an abbreviation of the author's name for second and subsequent citations, where the abbreviation is introduced in the first citation. It can also be used to "trick" the system in difficult cases.

lastchecked The date on which a web page has last been viewed.

original* The APA manual prescribes that one should always refer to the edition that is actually used. So if one uses a translation or an unchanged reprint of an earlier published work, the translation or reprint should be referenced. However, in these cases, publication details of the original work should be mentioned as well. An example of this was the work by Freud mentioned in section 1. Therefore, apacite supports several fields in which this information can be given. These fields are originaladdress, originalbooktitle, originaledition, originaleditor, originaljournal, originalnumber, originalpages, originalpublisher, originalvolume, and originalyear. These fields have the same meaning as their counterparts without the "original" prefix, except that they refer to the book or journal in which the work was originally published.

BibTeX also has a crossref field and it would be more elegant if this could be used to link the original publication to the cited work, but this does not work well. BibTeX uses this field to copy missing fields in the current entry from the crossreferenced entry. This is done before the information is handed over to `apacite.bst`, so it is not possible to capture this information. This feature is primarily intended for articles in a book, so that the editor, book title, address, publisher, and possibly other details of the book can be copied from the entry for the whole book. However, if both the translation and the original work are a book, each with a different publisher, the publisher field of the translation is not empty and thus the publisher field of the original work is not copied to the translation. Moreover, even if such information is copied, then `apacite.bst` would not be able to assess whether it belongs to the translation or to the original work. Finally, note that if two or more articles from the same book (i.e.,

with the same crossref field) are cited, BibTeX includes the whole book as an additional item in the reference list. This is not allowed by the APA manual, so the crossref field should not be used.

The amsrefs package solves these problems by allowing nested fields (within a LaTeX, not a BibTeX, context), called *compound* fields, e.g., a translation field that contains all the information (journal, volume, pages, etc.) of a complete entry as subfields.

symposium The name of the symposium or meeting at which a lecture was given.

text Used for items of type `@literal`. This field contains the complete literal text to be used in the reference list.

translator The translator of a book or article.

url The url (web address) of a web page.

6 Customization

The default behavior of apacite is intended to adhere to the rules of the APA manual as strictly as possible. There are, however, many journals, universities, and publishers that define their own citation styles, which are very similar to the APA style. Therefore, it is convenient that relatively small and/or straightforward changes to the default behavior can be easily made. apacite supports this in two ways: It provides several options to change parts of its behavior, and it defines most formatting characteristics through commands, which can be redefined by users in their own style files or before `\begin{document}` in their `.tex` files.

Apart from a couple of technical options, there are options to turn author indexing on and off and to control the formatting of the author index (see section 8), to choose whether the reference list should be put in a section or in a chapter, to choose whether the reference list section or chapter should be numbered or not, whether the reference list must be included in the table of contents or not, and whether the reference list should be started on a new page.

Most of the punctuation used in the citations and reference list is implemented through LaTeX commands instead of explicit symbols. Consequently, the user can fine-tune the behavior of apacite by redefining these commands, through `\renewcommand` after apacite has been loaded. For example, the opening parenthesis of a citation is defined in the command `\BBOP` and the closing parenthesis is defined in the command `\BBCP`. The default values are '(' and ')', but by including the commands

```
\renewcommand{\BBOP}{[}
\renewcommand{\BBCP}{]}
```

after loading apacite, a typical citation will look like '[Jones, 2001]' instead of the default '(Jones, 2001)'.

An example of a customizable command that changes the formatting behavior of apacite is `\bibliographytypesize`, which defines the size of the font to be used for the reference list. By default, it is `\normalsize`, but for Wansbeek and Meijer (2000), I defined it as `\small` (and even then the reference list took up 34 pages). Other examples are `\bibleftmargin` and `\bibindent`, which define the indentation of the entries in the reference list, and `\bibitemsep`, which defines the amount of vertical white space between entries in the reference list.

There are many specific pieces of text that are put into the reference list or a citation by apacite. Examples are "et al.", "in press", "and", and "Tech. Rep.". Almost all of these are to some extent language-specific, and sometimes style-specific even within the same language. Therefore, they are implemented through LaTeX commands, so that users can easily change them.

Experimental versions of apacite (2006 and later) have more rigorously shifted formatting from BibTeX to LaTeX. A still relatively simple example of this (capitalization of titles) was given in section 3.2 above.

7 Language support

The APA is, of course, American, and therefore the rules in the APA manual are also based on the (U.S.) English language. Because apacite was primarily designed to implement the APA rules, apacite originally did not contain explicit support for other languages. However, as mentioned above, most language-specific elements have been implemented in the form of LaTeX commands, so that users could define their own LaTeX package in which these elements were changed. This widens the applicability of apacite considerably. Many journals in non-English languages and universities in countries where other languages are spoken base their rules on the APA manual. Therefore, it is efficient that, with a few adaptations, apacite can also be used in these circumstances. Because of its importance, since 2005, apacite has contained some explicit language support, so that users do not have to make their own adaptations.

The apacite distribution now contains files that have names according to the construction ⟨*language*⟩`.apc`, i.e., `english.apc`, `dutch.apc`, etc. These files contain the language-specific modifications of apacite, mainly translations of terms like "and", "Ed.", etc., and have been written by apacite users that are (native) speakers of the languages involved. For example, `apacite.sty` contains the line

```
\newcommand{\BBAB}{and} % between authors in text
```

and `dutch.apc` redefines this as

```
\renewcommand{\BBAB}{en}
```

If the babel, german, or ngerman package is loaded, apacite is able to determine the language of the document that is processed. Then the corresponding `.apc` file, when available, is read. In this way, language-specific elements are changed to the relevant language. This is done fully automatically, the user does not have to do anything explicitly. Sometimes an `.apc` file makes some assumptions (such as `greek.apc`, which assumes ISO-8859-7 encoding) or makes some choices that are nontrivial. Of course, when the definitions in the relevant `.apc` file are different from the desired ones, users can redefine them again after loading apacite.

One file is defined for each "language", which can be used for several "dialects" (in babel terminology). For example, `english.apc` is also used if the language is "american". See the documentation of the babel and (n)german packages for a list of dialects of the language files that are recognized. The language files that are currently (July 2007) available are: `dutch.apc`, `english.apc`, `german.apc`, `greek.apc`, `ngerman.apc`, `norsk.apc`, `spanish.apc`, and `swedish.apc`. If an `.apc` file is not available for the required language, users can use one of the supplied ones as a template and write their own. These will be included in a next version of apacite when they are sent to me.

Specifically, apacite tries to determine the relevant language file as follows: The abovementioned packages keep a numbered list of loaded languages, with 0 (zero) being the default language, typically English. The name of a language is mapped to its number by the command `\l@`⟨*language*⟩, so typically `\l@english` is 0, but also `\l@american` is 0 if the "american" language is loaded. The number of the currently active language is analogously stored in the command `\language`. Furthermore, the name of the currently active language is stored in the command `\languagename`. Now, apacite checks at `\begin{document}` which language is active. Then, it is checked whether the file `\languagename.apc` can be found by TEX. If not, it is checked whether `\language` is equal to `\l@`⟨*language*⟩ for one of the languages that are supplied in the apacite distribution, which should cover "dialects" of known languages.

The language support in apacite is still very limited. There are many aspects that are not yet covered, like different forms of dates (month-day vs. day-month). The babelbib package offers very sophisticated support of different languages in the reference list and a future version of apacite may use some of its features.

8 Generating an author index

The apacite package contains an option to (almost) automatically generate an author index. In this case, the apacite package automatically loads the index package that supports multiple indexes, so that one can have a subject index as well as an author index. In `apacite.sty`, an author index is defined, which has the identifier string "`autx`". The `\indexentry` commands for *MakeIndex* are

written to an `.adx` file and the formatted index should be written to an `.and` file by *MakeIndex*. The heading of the author index is defined in the command `\authorindexname`, which can be redefined by the user.

To use the author index option, the `apacitex.bst` BibTeX style file must be requested in the `\bibliographystyle` command. The author index is then included by putting

```
\printindex[autx]
```

at the point in the LaTeX document where the index is supposed to appear. Correct generation of the author index is done by the *MakeIndex* program and generally requires a few additional LaTeX runs.

The apacite package provides options to change the appearance of the index somewhat. With the index option, the indexing facility is turned on, but the `theindex` environment is not explicitly (re)defined by apacite. The stdindex option explicitly uses the definition of the `theindex` environment that is defined in the index package [1995/09/28]. With this definition, the index does not appear in the table of contents. With the tocindex option, this definition is augmented with a table of contents entry. Finally, with the emindex option, there are some alternative definitions. It writes a table of contents entry as well, but the index itself is now set in two columns using the multicol package instead of the `\twocolumn` command, the text of the index is set in small type, and the page head is not put in uppercase.

A small example document (`doc3.tex`) that uses the indexing facility is the following:

```
\documentclass{article}
\usepackage[emindex]{apacite}
\begin{document}
\citeA{MaBaMc1988} found that fit indexes may be unreliable.
But some are better than others \cite{MaBaMc1988}.
\bibliographystyle{apacitex}
\bibliography{db1}
\printindex[autx]
\end{document}
```

It differs only slightly from the earlier example file `doc2.tex`. The only differences are the emindex option of the apacite package, the apacitex bibliography style instead of apacite, and the inclusion of the `\printindex` command.

The corresponding BibTeX run generates the file `doc3.bbl`, with (a.o.) the following contents:

```
\bibitem[\protect\citeauthoryear{Marsh\AX{marsh  hw @Marsh, H.~W.},
   Balla\AX{balla  jr @Balla, J.~R.}, \BBA{}
   McDonald\AX{mcdonald  rp @McDonald, R.~P.}}{%
   Marsh\AX{marsh  hw @Marsh, H.~W.} et al.}{1988}]{MaBaMc1988}
Marsh, H.~W.\AX{marsh  hw @Marsh, H.~W.},
Balla, J.~R.\AX{balla  jr @Balla, J.~R.}, \&
McDonald, R.~P.\AX{mcdonald  rp @McDonald, R.~P.}
```

This shows that the connection between the `.bbl` file and the author index is through `\AX` commands that are included in the `.bbl` file. These are responsible for the author indexing facilities. `\AX` is essentially the same as `\index[autx]`. The example also illustrates that apacite uses the ⟨*key*⟩`@`⟨*visual*⟩ method of indexing, where ⟨*key*⟩ indicates how the item should be sorted, whereas ⟨*visual*⟩ indicates how it should be formatted in the index. This method solves some problems (found in earlier versions of apacite) with accents, lower case letters, and braces, which led to incorrect sorting of the index. Like the sorting key of the reference list the ⟨*key*⟩ here is in lower case, with all accents and braces removed.

After the third LaTeX run, the file `doc3.aux` contains the following lines

```
\bibcite{MaBaMc1988}{\citeauthoryear{Marsh\AX{marsh hw @Marsh, H.~W.},
   Balla\AX{balla jr @Balla, J.~R.}, \BBA{}
   McDonald\AX{mcdonald rp @McDonald, R.~P.}}{Marsh\AX{marsh hw @Marsh, H.~W.}
   et al.}{1988}}
\@writefile{autx}{\indexentry{marsh hw @Marsh, H.~W.}{1}}
```

The `\@writefile` command writes the indexing information to the `.adx` file. The result is that after the third LaTeX run, the file `doc3.adx` contains several lines of the form

```
\indexentry{marsh hw @Marsh, H.~W.}{1}
```

After the third LaTeX run, which gives the final formatted document if the indexing facility is not used, the *MakeIndex* program must be invoked to generate the author index. This is done by a (shell) command like

```
makeindex -o doc3.and doc3.adx
```

or by clicking on the corresponding options in a GUI-based TeX distribution. *MakeIndex* then reads the indexing information from `doc3.adx` and writes the formatted index to `doc3.and`. The latter file has the following contents:

```
\begin{theindex}

  \item Balla, J.~R., 1

  \indexspace

  \item Marsh, H.~W., 1
  \item McDonald, R.~P., 1

\end{theindex}
```

Like most files discussed here, this is a LaTeX file. It can be edited when desired. In the subsequent (fourth) LaTeX run, this file is processed by LaTeX at the position of the `\printindex` command. After the fourth LaTeX run, the typeset document is complete. Thus, with author indexing, the following sequence of programs is run: (1) LaTeX, (2) BibTeX, (3) LaTeX, (4) LaTeX, (5) *MakeIndex*, (6) LaTeX. With very complex documents, like the apacite manual, additional

Marsh, Balla, and McDonald (1988) found that fit indexes may be unreliable. But some are better than others (Marsh et al., 1988).

References

Marsh, H. W., Balla, J. R., & McDonald, R. P. (1988). Goodness-of-fit indexes in confirmatory factor analysis: The effect of sample size. *Psychological Bulletin*, *103*, 391–410.

Author Index

Balla, J. R., 1
Marsh, H. W., 1
McDonald, R. P., 1

Figure 3: Typeset document using bibliography style `apacitex` and using the apacite LaTeX package with option emindex.

LaTeX runs may be necessary to resolve all cross-references, and *MakeIndex* may have to be invoked twice.

The resulting typeset document is shown in Figure 3.

The author indexing part works very well for most commonly encountered cases. However, it does not work (entirely) correctly if special measures need to be taken to get them formatted correctly in BibTeX (citations, reference list). Examples from the APA manual that lead to undesirable or incorrectly formatted entries in the author index are rare types of "authors" as in "Bender, J. (Director)." and "Bulatao, E. (with Winford, C. A.).". These problems are caused by the "tricks" that are necessary to make BibTeX format them correctly in the reference list.

An easy solution to incorrect formatting in the author index is to edit the `.bbl` file manually. This should be done at a time when BibTeX will not have to be run again. Otherwise, the changes would be overwritten by the next BibTeX run. This editing of the `.bbl` file is not in the LaTeX-spirit, but in my experience — I used it for two books with lots of references (Meijer, 1998; Wansbeek & Meijer, 2000), with a previous version of apacite that caused many more problematic cases — this takes very little time, usually only minutes (compare that with the time spent on writing a book, or with the time that would be necessary to manually make an author index). The correct author index could also be obtained by editing the final `.and` file, which contains the final formatted author index. However, it is more convenient to edit the `.bbl` file, which is usually "final" at a much earlier stage.

Just as with the sorting of the reference list, the index may not be sorted as desired for certain languages if there are accents or other "special" charac-

ters. Different scripts may not be handled entirely correctly either. These are problems out of the reach of apacite.

9 Compatibility

There are two kinds of compatibility that can be distinguished. The first is that input files that can be used with apacite can also be used with other packages, so that one can quickly switch from one citation style to another. In particular, this means that the same reference types and fields can be used in the `.bib` file and these should have the same meaning, and that the same citation commands can be used in the `.tex` file and these should have the same meaning as well. The second kind of compatibility is that apacite can be used together with other packages or programs without errors or undesirable side-effects.

The first kind is very important. If one has to write a completely different `.bib` file for different citation styles, there does not seem to be an advantage in using BibTeX. The reference list may just as well be written explicitly in LaTeX then. However, not even this point can be achieved completely, although the vast majority of the items will be the same for different styles. But there remain a number of choices that are style-specific and that lead to differences in the `.bib` file. Examples from the APA manual are:

- If a referenced book is volume III according to its title page, this should be referenced as Vol. 3 according to the APA manual rules, but that may not be the case with other styles.
- Publisher names should be abbreviated according to the APA manual, e.g., "John Wiley & Sons" becomes "Wiley". Other styles do not do this.
- Yearbooks like *Annual Review of Psychology* should be treated as journals according to the APA rules, whereas other styles treat these as books.

Of course, the additional fields (such as translator and the original* fields) that are used by apacite but are not defined in other BibTeX styles are apacite-specific. These fields are ignored by other citation styles, which usually means that such styles do not require this information and the work is formatted correctly. Analogously, the apacite-specific reference types such as `@lecture` are not explicitly supported by other styles. These will treat such references as being of type `@misc`. Whether this results in a satisfactory formatting of the reference varies and depends primarily on the amount of information in the entry that is ignored.

Thus, not all entries in a `.bib` file that is designed to be used with apacite will be suitable for some other citation styles. On the other hand, most citation styles and journals are not as critical as APA journals and many styles in psychology and other social sciences (including economics) are very similar, so a `.bib` file that is tailor-made for apacite is likely to be suitable enough for the styles of most relevant alternative journals.

The situation for `.tex` files is less positive. The apacite citation commands are directly based on those of its immediate predecessor, theapa. But the use of '<' and '>' for prefixes is not used by other packages. The influential natbib package uses `\citep` and `\citet` instead of `\cite` and `\citeA`, and uses many more alternative commands. The chicago package uses `\citeA` instead of `\citeauthor`, many "numerical" citation styles only recognize `\cite`, and so forth. Because the different citation packages use different commands, or use the same commands with different meanings, it is difficult to ensure that the same `.tex` file can be used with different citation packages.

The second kind of compatibility is also very important, especially with other packages that are complementary to apacite in some sense. It would be very annoying having to choose between using babel and apacite, or between hyperref and apacite, etc. These packages serve totally different purposes, cannot be compared usefully, and are more valuable when they can be used jointly.

Up to 2005, there have been some compatibility problems between apacite and some other packages, most notably hyperref and several packages that support multiple bibliographies (bibtopic, bibunits, multibbl, multibib, splitbib). These problems have now been largely solved, although a few issues are still unresolved.

Most programs for converting LaTeX to other formats (rtf, html) do not support apacite satisfactorily as well. It appears that the TeX4ht converter currently works best. This largely operates on a lower-level TeX and uses LaTeX to translate macros (commands) to such lower-level TeX commands. Thus, it supports package- and user-defined LaTeX commands in principle. Other converters must include code for every LaTeX command. Therefore, they are not able to handle user-defined changes and they support package-defined commands only if these are also programmed in the converter. This is less flexible and leads to more problems.

The natbib package is a general purpose citation package that is intended to work with a broad range of BibTeX (and non-BibTeX) styles that generate the bibliography. The natbib package is quite advanced and can be used to switch easily between completely different citation styles. Furthermore, the apparent popularity of natbib has inspired writers of packages that would otherwise be incompatible with natbib to write code to resolve these incompatibilities. Using natbib for the citations has some advantages over using apacite for the citations. The most important ones are natbib's option of sorting citations within a single citation command and natbib's better compatibility with other packages. However, natbib does not fully comply with the APA rules. Therefore, for submission to APA journals, it is necessary to use apacite for the citations. Because of the leading role of natbib, future versions of apacite will support the natbib citation commands (with APA-style formatting), but the current version does not do that.

10 The future of apacite

The apacite package is maintained fairly actively. This means that I tend to respond quickly to user questions, requests, and reports of apparent bugs. Furthermore, new features are added to apacite frequently. This has resulted in new releases that are irregularly issued. The frequency depends on how important it is to release an update soon and how much time I am able to spend on it, because the development of apacite has never been one of the main goals of my employers. I intend to remain actively engaged in the development of apacite, although I will have to do this mostly in my own (spare) time.

With any kind of software, there is usually a list with known problems (bugs) and desirable future work (to-do), and apacite is no exception. Some issues have already been mentioned above, and the documentation of apacite also gives a list. The number of known problems is fortunately very small. These problems pertain mainly to rare instances of incorrect formatting and to a few incompatibilities with other packages. Solving these problems has the highest priority in the development of apacite. Other developments with high priority are supporting the natbib citation commands (`\citet`, `\citep`, etc.) and improving and extending the language support, possibly by connecting with babelbib.

Another direction of development, already mentioned earlier, is increasing the customizability of the reference list. As stated above, BibTeX does not explicitly support options, and therefore, customizability takes the shape of writing flexible LaTeX macros to the `.bbl` file instead of fixed formatting. As mentioned in section 6, many fixed strings and punctuation elements, have been implemented through such LaTeX commands, which can be redefined by the user. In experimental versions of apacite, this direction has been followed further, and future versions of apacite will continue to be developed in this direction. For example, the `.bbl` file contains information in the form

```
\APACjournalVolNumPages{Journal of Applied Psychology}{78}{}{443--449}
```

The LaTeX command `\APACjournalVolNumPages`, which has four mandatory arguments (journal, volume, number, pages) does the actual formatting of this information. This gives much more freedom for customization of the reference list. Probably, this will lead to a number of additional options to `apacite.sty`, but the user (or an external LaTeX package) can also completely redefine commands like `\APACjournalVolNumPages`, thereby having complete freedom over its formatting. However, this setup still limits the formatting of the reference list, because, e.g., the year cannot be inserted after the journal. The amsrefs package does all of its formatting in LaTeX and thus does allow such far-reaching changes. Whether apacite will eventually move in that direction is uncertain. The formatting and the sorting are to some extent dependent on each other and the sorting is not done in LaTeX. Presently, I believe that some tasks are better left to BibTeX.

The main tasks that must still be done in BibTeX, by `apacite.bst`, are sorting the reference list, generating the citation labels, and making some choices regarding the formatting of names (authors, editors). These three tasks are so closely related that they cannot be well separated. It would be desirable to provide several options for these tasks as well, e.g., generating an unsorted reference list or one sorted in chronological order (for CV's), giving full given names instead of initials, and putting the initials or full given names in front of the surnames (possibly with the exception of the first author). These are a few examples of requests that I have received. Although BibTeX does not explicitly support options, it is still possible to use options by citing pseudo-references from which the desired behavior can be extracted. Consider, for example, citing the following entry through `\nocite{[unsorted]}`:

```
@bst-option{[unsorted],
}
```

The `.bst` file can be written in such a way that it first checks whether there are references of entry type `@bst-option`, and then depending on the value of the citation key (`cite$`), here `[unsorted]`, changes its behavior. I have had this idea for a couple of years now and later found out that it had already been implemented in old versions of the `amsxport.bst` file of the amsrefs package. It has not yet been implemented in apacite, because Oren Patashnik has announced a new version of BibTeX (version 1.0) in which options would be explicitly supported. However, no signs of its actual development have been received, so future versions of apacite may contain options in the way described here.

Finally, although I have my own agenda, suggestions and requests by users will be taken into account as well and may change priorities.

Acknowledgements

The author would like to thank Athanassios Protopapas and Ioannis Dimakos for useful comments on an earlier version of this paper.

References

American Psychological Association. (1984). *Publication Manual of the American Psychological Association* (3rd ed.). Washington, DC: Author. (with revisions)

American Psychological Association. (2001). *Publication Manual of the American Psychological Association* (5th ed.). Washington, DC: Author.

Kopka, H., & Daly, P. W. (2004). *Guide to LaTeX* (4th ed.). Boston, MA: Addison-Wesley.

Lamport, L. (1994). *LaTeX: A document preparation system. User's guide and reference manual* (2nd ed.). Reading, MA: Addison-Wesley.

Meijer, E. (1998). *Structural equation models for nonnormal data*. Leiden, The Netherlands: DSWO Press.

Mittelbach, F. & Goossens, M. (2004). *The LaTeX companion* (2nd ed.). Boston, MA: Addison-Wesley.

Wansbeek, T., & Meijer, E. (2000). *Measurement error and latent variables in econometrics*. Amsterdam: North-Holland.