

# The **apacite** package\*

Citation and reference list with  $\text{\LaTeX}$  and  $\text{\BibTeX}$  according to the rules of the  
American Psychological Association

Erik Meijer  
apacite at gmail.com

2009/03/01

## Abstract

This document describes and tests the **apacite** package [2009/03/01]. This is a package that can be used with  $\text{\LaTeX}$  and  $\text{\BibTeX}$  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. The package can be customized in many ways.

---

\*This document describes **apacite** version v4.05 dated 2009/03/01.

## Contents

# 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 (?, ?). In the sequel, 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. L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X (?, ?, p. 7).

An implementation of the formatting rules of the APA manual for use with L<sup>A</sup>T<sub>E</sub>X is the `apa` class of Athanassios Protopapas. This handles all kinds of issues about general document formatting, title page, section headings, figures and tables, and so forth. Therefore, if you intend to submit a manuscript to an APA journal, I strongly recommend using the `apa` class.

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. The `apa` class requires the `apacite` package, but `apacite` can be used without `apa`. The current document, for example, does not use the `apa` class, because I find it desirable that possible users can study the `apacite` package and its documentation without having to install several other classes and packages first. Therefore, the current document uses standard L<sup>A</sup>T<sub>E</sub>X as much as possible.

## Some history

Before the advent of the first version of `apacite` in 1994, several authors have written T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X packages (`.sty`) and B<sub>I</sub>B<sub>T</sub><sub>E</sub>X style files (`.bst`) with suggestive names as `apalike`, `newapa`, and `theapa` to try to capture some of the requirements of the APA manual. A severe problem with the APA citations is that, if the number of authors is between three and five, inclusive, the first citation must include all authors, whereas subsequent citations should refer to the first author “et al.”. This problem had been solved by the `newapa` and `theapa` packages.

A lot of APA peculiarities, however, were not covered by these citation styles. Examples are:

- In the reference list, authors should be formatted with their initials after their surnames. Editors, however, should be formatted with their initials *before* their surnames if they are not in the author position.
- 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.
- Edition and volume of a book and the pages of an article in that book should be gathered between a single pair of parentheses.
- The publisher of a book and its address should be given as “address: publisher”, and not as “publisher, address”.

- When the work used is a reprint or translation of an earlier work, this should be mentioned in the reference list. In text, the year must be given in the form “1923/1961”.
- There are several “tricky” things with brackets and parentheses, for example, with the type of an article (e.g., “Letter to the editor”, “Review of the book *The social life of information*”, “Abstract”, “Special issue”). Furthermore, magazine and newspaper articles are treated differently than journal articles; papers presented at meetings must be treated differently; translators of articles and books and chairs of symposia must be introduced; an article can have editors instead of authors (e.g., with a special issue); a PhD thesis can be abstracted in *Dissertation Abstracts International*; a (technical) report of some institution can be an (edited) collection of articles.

The `apacite` package is devised to get (much) closer to the APA rules than the abovementioned other packages and styles.

The original version of the `apacite.sty` L<sup>A</sup>T<sub>E</sub>X package consisted for a large part of the citation part of `theapa.sty`. The current version, however, has been changed and extended in many ways and can therefore not 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.

Although the `apacite.bst` BibT<sub>E</sub>X style started out as a slightly adapted version of `theapa.bst`, the number of changes became so large and things became so unmanageable that I decided to write a completely new bibstyle, although using some small parts of the original. The `apacite.bst` file can therefore not anymore be seen as an adaptation of `theapa.bst`. Due to the complex rules of the APA manual, it has become a large file. In the original process of writing (back in 1994), the error

`You've exceeded BibTeX's wizard-defined function space`

was very hard to overcome. It appears that later versions of BibT<sub>E</sub>X allowed much more space, because I have not received this message anymore in recent years, although the bibstyle has been expanded considerably. However, if you run an old BibT<sub>E</sub>X, you may encounter this (and similar) error messages.

In the “dark years” between 1994 and 2003, in which there was no official new version of `apacite`, I have been fixing small bugs, discovered either by me or by users who sent in their bug reports, and adding some small options that were easy to incorporate and that I found useful or that were requested by a user. Furthermore, in 1998, I spent considerable effort in adding an (almost) automatic author indexing option, first used for my PhD thesis (?, ?).

In the first few months of 2003, I fixed a number of problems with the compatibility between `apacite` and some other important packages. Although I had been planning (or wishing, rather) to release a proper official update of `apacite` for some time, I had not done so, mainly because I had not yet taken the time to update the documentation. The compatibility issues were so severe, however, that I decided in March 2003 to release an intermediate update, which thus became the first official release since the original version. This version contained a lot of files with compatibility patches and tests. The main shortcomings of that version were the lack of up-to-date documentation and the lack of agreement with the latest version of the APA manual (5th ed., 2001).

In September 2003, I finally released a complete, fully updated and significantly extended version of `apacite`, including extensive documentation. However, that version still was not able to format some citations correctly. The [2004/07/01] version was a small improvement of the [2003/09/05] version, but fixed the important inadequacies of previous versions.

The [2005/06/08] version introduced (limited) support of non-English languages. This is clearly a non-APA direction, because all APA publications are in English. However, many journals, universities, etc. in non-English speaking countries base their rules on the APA rules, and thus it may be useful to have the possibility to use **apacite** in combination with documents in non-English languages as well. This language feature has been further extended in the [2007/09/03] version. See section ?? for a description of the implementation of this feature.

The most salient change in the [2007/09/03] version was the shift of many formatting decisions from BibTeX to L<sup>A</sup>T<sub>E</sub>X, so that they are easier to change by the user.

The current version adds some further improvements in customizability and language support, and adds support for annotated bibliographies.

Apart from the language support and the facilities for customization, the current version of **apacite** is aimed at conforming with the 5th edition of the APA manual. It remains, however, imperfect, although the instances of incorrectly formatted cases will be rare. The problems can be solved by “tricking” the style (e.g., by defining the **number** field as “2, Serial No. 186” if the issue number and serial number of a journal are both important), but this makes the **.bib** file less compatible with other styles. In the test examples in section ?? below, many tricks are used and discussed. Remaining problems, omissions, and things-to-be-done can be found in section ??.

## Philosophy of **apacite**

As can be deduced from the discussion thus far, the first priority of **apacite** is to implement the rules of the APA manual with regard to citation and reference list as closely as possible. However, just like its predecessors, and actually expanding much beyond their realm, **apacite** offers many possibilities for customization as well. Many details of **apacite**, particularly punctuation and some fixed texts (e.g., “Tech. Rep.”) can be changed easily by the user by redefining some commands in L<sup>A</sup>T<sub>E</sub>X. Furthermore, **apacite** also offers several proper options to change some of its settings.

Whether certain options or customizable aspects are implemented depends on two criteria: (1) Is it possible, easy (enough), and convenient to implement them without compromising the ability to adhere to the APA rules, and (2) Do I (EM) consider them important or useful enough to spend time to implement them. Actually, the decision process is the reverse of this: First, I decide whether I find it a relevant or useful option. If not, I will not implement it. If so, I will think about if and how I can implement it. If I have an idea for a solution that is practically feasible, I will pursue it. If I don’t see a solution, if I think it will take me too much time, or if I think a solution will be inconvenient to other users, then I will not pursue it.

## New in this version

The changes since the [2007/09/03] version are:

- Addition of **french.apc** to the list of language-specific files.
- Improvements in various **.apc** files. In particular, in addition to some small changes in various files, all of the **.apc** files are greatly extended, supporting more language-specific elements.
- Added support for annotated bibliographies through the **apacann.bst** and **apacannx.bst** BibTeX style files. See section ?? for details.
- Introduction of the **\APACstdindex**, **\APACtocindex**, **\APACemindex**, and **\APACltxemindex** commands, which change the style of the index.

- Improved source code documentation in the `.dtx` file.
- Some minor changes and bug fixes. In particular, a few problems that showed up when the reference list contains multiple works by the same author(s) published in the same year, have been resolved.

## About this document

The current document describes how to use **apacite** and largely assumes knowledge of or access to the standard BibTeX documentation, such as ? (?), ? (? , chap. 12), or ? (? , chap. 12 and 13). Hence, this document does not always describe how to use some of the citation commands or how to construct a bibliography database file in detail if there is no **apacite**-specific element to it. This document comes in two versions. The version supplied with the distribution is the user's manual. As is customary with packages that are distributed in `.dtx` form, it is also possible to regenerate the user's manual in such a way that it includes the documented source code of the package as well. This is currently still in a primitive form compared to other packages, but it will be improved with later releases. The **README** file describes how the documented source code can be included in this manual. ? (?) contains some simple examples in which **apacite** is compared to standard LaTeX/BibTeX citation, as well as a description of how **apacite** works technically.

## 2 Installation, package loading, and running BibTeX

**apacite** is distributed as a `.dtx` file, like most LaTeX packages. The file **apacite.dtx** is supplemented by a **README** file, which gives a brief introduction and installation instructions, the user's manual in the file **apacite.pdf** (which you are reading right now), and the installation file **apacite.ins**. Strictly speaking, only the **apacite.dtx** file is necessary, because the installation file is regenerated from it if it is not available, and the user's manual is generated by running LaTeX on **apacite.dtx**. But it is customary (and convenient for potential users) to include the other files as well.

The LaTeX packages, BibTeX style files, and other files in the distribution are generated by running LaTeX on **apacite.ins**. This generates the following files:

**apacite.sty** The LaTeX citation package. This must be placed in a directory where TeX can find it.

**apacite.bst** The BibTeX reference list style. This must be placed in a directory where BibTeX can find it.

**apacitex.bst** The BibTeX reference list style with added author index support. This must also be placed in a directory where BibTeX can find it.

**apacann.bst**

**apacannx.bst** Versions of **apacite.bst** and **apacitex.bst** that generate annotated bibliographies (if the **annote** and/or **annotate** fields are provided). Again, these must be placed in a directory where BibTeX can find it. Most probably, some time in the future, the four `.bst` files will be combined in **apacite.bst**, with the desired behavior induced by options, but this is nontrivial and currently not implemented yet.

**apa5ex.bib** The file with bibliographic information about the references used in the user's manual (i.e., the current document). This is also useful for users to find out how certain nontrivial problems can be solved. This can best be kept in the same directory as **apacite.dtx**.

**apacite.drv** Documentation driver. Run  $\text{\LaTeX}$  on this file to regenerate the user's manual. This is the current document, and can also be obtained by running  $\text{\LaTeX}$  on **apacite.dtx**, but you can edit the file **apacite.drv** to change some settings, e.g., choose whether or not the documented source code must be included in the manual or not. Please don't edit **apacite.dtx** itself. The file **apacite.drv** can best be kept in the same directory as **apacite.dtx**.

**\*.apc** Language-specific modifications of **apacite**. See section ?? for a discussion of these. These must be placed in a directory where  $\text{\LaTeX}$  can find them. The files that are currently supplied are **english.apc**, **dutch.apc**, **finnish.apc**, **french.apc**, **german.apc**, **ngerman.apc**, **greek.apc**, **norsk.apc**, **spanish.apc**, and **swedish.apc**.

**apacdoc.sty** A  $\text{\LaTeX}$  package that contains commands and settings used in this user's manual. This can be placed in a directory where  $\text{\TeX}$  can find it, but given that it is primarily useful for processing the user's manual and not intended as a package for wider usage, it can also be left in the same directory as **apacite.dtx**. See section ?? for a further discussion of this package.

Previous versions of **apacite** provided the file **apacite.tex**, which contained the user's manual, but this is now included in **apacite.dtx** and can be formatted by running  $\text{\LaTeX}$  on **apacite.drv**. The **apacite.sty**  $\text{\LaTeX}$  package is loaded by putting

```
\usepackage[<options>]{apacite}
```

somewhere in your document between `\documentclass` and `\begin{document}`, or putting

```
\RequirePackage[<options>]{apacite}
```

in your own personal  $\text{\LaTeX}$  package (say, **mysettings.sty**) that is loaded by your document.

To load the **apacite.bst** or **apacitex.bst** bibliography style in  $\text{\BibTeX}$ , put

```
\bibliographystyle{apacite}
```

or

```
\bibliographystyle{apacitex}
```

respectively, in your document before the `\bibliography` command. The position of the bibliography (reference list) is determined by the line

```
\bibliography{<bibfiles>}
```

where *<bibfiles>* is a list of filenames with **.bib** extension, which contain the bibliographic information that is used by  $\text{\BibTeX}$  to construct the reference list. Usually, the `\bibliographystyle` and `\bibliography` are kept together (immediately follow each other) in the document, although when you are using the **apa** document class, **apacite** is used by definition and you are not supposed to use the `\bibliographystyle` command yourself. See the documentation of the **apa** documentclass for details about this.

If you use one of the author indexing options, the author index is put in the L<sup>A</sup>T<sub>E</sub>X output by the line

```
\printindex[autx]
```

If you put this line in your document, but don't use one of the author indexing options, it will be ignored. For more on author indexing, see section ??.

To get all parts in the final output, the following sequence of runs should typically be taken (when starting from scratch): (1) L<sup>A</sup>T<sub>E</sub>X, (2) B<sub>I</sub>B<sub>T</sub><sub>E</sub>X, (3) L<sup>A</sup>T<sub>E</sub>X, (4) L<sup>A</sup>T<sub>E</sub>X, and, when author indexing is on, (5) *MakeIndex*, (6) L<sup>A</sup>T<sub>E</sub>X, and (7) L<sup>A</sup>T<sub>E</sub>X. The last one is to get the index in the table of contents. If the table of contents is on a regular page, i.e., an arabic-numbered page instead of a roman-numbered page in the front matter, it may even be necessary to run *MakeIndex* another time, followed by L<sup>A</sup>T<sub>E</sub>X once or twice. Occasionally, somewhere in the process, L<sup>A</sup>T<sub>E</sub>X may complain about labels that may have changed, which requires even more additional L<sup>A</sup>T<sub>E</sub>X runs at that stage. So the number of runs that are necessary to get everything right may become large.

### 3 Package options

The following options are recognized by **apacite**:

**BCAY**

This is a technical option for backwards compatibility with old versions (pre-[2003/09/05]) of **apacite**. In those versions of **apacite**, the `\BCAY` construction was used to pass relevant citation information from the `.bbl` file (B<sub>I</sub>B<sub>T</sub><sub>E</sub>X's output) to L<sup>A</sup>T<sub>E</sub>X. This was taken over from its immediate predecessor, Young U. Ryu's **theapa**. However, **natbib** does not recognize the `\BCAY` construction, but it does recognize the analogous `\citeauthoryear` construction, which was also used by an earlier predecessor of **apacite**, **newapa**. Therefore, **apacite** has reverted to `\citeauthoryear` as well. This makes different versions of **apacite** incompatible with each other, because it is not possible to support both constructions at the same time. This option is used to fix that: In the (unlikely) event that you *must* use a `.bbl` file that is generated by an old version of **apacite**, you can turn this option on.

**index**

**indexpackage**

**noindexpackage** The **index** option turns author indexing on. See section ?? for a discussion of the author indexing facility. The **index** option should be used with the **apacitex.bst** (or **apacannx.bst**) B<sub>I</sub>B<sub>T</sub><sub>E</sub>X style, although it does not give errors with **apacite.bst** (or **apacann.bst**), but simply does not give author index entries, so then this option typically does not have any effect (and an undesirable effect if it does). The **indexpackage** and **noindexpackage** select whether the author index entries should be generated according to the method of the **index** package or using a more standard-L<sup>A</sup>T<sub>E</sub>X method.

By default, if the **index** option is requested, the **indexpackage** option is turned on as well, consistent with the behavior of previous versions of **apacite**.

**noindex**

Turns author indexing off (the default). Typically used with **apacite.bst**, but can also be used with **apacitex.bst**. In the latter case, the author indexing commands are simply



ignored. Therefore, `apacite.bst` is actually superfluous, but because author indexing will be used rarely and `apacitex.bst` is more likely to lead to errors or incompatibilities, a “clean” (no author indexing) version, `apacite.bst`, is provided as well.

**suppresscorporate**

Excludes corporate authors from the author index. The `\bibcorporate` command must be used in the `.bib` file to denote a corporate author; see sections ??, ??, and ??.

**includecorporate**

Includes corporate authors in the author index, provided that an author index is requested of course.

**stdindex**

**tocindex**

**emindex**

**ltxemindex**

These options select the style of index formatting. They all imply the `index` option. The first three of these imply the `indexpackage` option, whereas the fourth implies `noindexpackage`. See section ??.

**numberedbib**

This option implies that the bibliography (reference list) is a numbered section or chapter, e.g., “6. References”, instead of just “References”.

**unnumberedbib**

The reverse of `numberedbib`: The bibliography is an unnumbered section or chapter. This is the default. However, it is possible that when using the `apa` document class, then `numberedbib` works better, because that class turns section numbering off anyway and it may be that `apa`’s page headings work well if the reference list is a `\section` and not if it is a `\section*`. I have not experimented with this (yet), however.

**sectionbib**

With this option, the bibliography is a section and not a chapter. Mainly useful in combination with the `chapterbib` package. Therefore, it will be discussed in more detail in section ??.

**nosectionbib**

With this option, the bibliography is a chapter, if the `\chapter` command is defined. Otherwise, it is always a section. Again, see section ??.

**tocbib**

This puts the bibliography in the table of contents, even if it is unnumbered, provided of course that a table of contents is requested in the document (by `\tableofcontents`). This is the default.

**notocbib**

This does not put the bibliography in the table of contents if it is an unnumbered section or chapter. If it’s numbered, it is always in the table of contents.

Table 1: Examples of usage of basic citation commands.

| Command   | Result                                 |
|---|--|
| <code>\cite&lt;e.g.,&gt;[p.~11]{Jone01,Ross87}</code>       | (e.g., Jones, 2001; Ross, 1987, p. 11) |
| <code>\citeNP&lt;e.g.,&gt;[p.~11]{Jone01,Ross87}</code>     | e.g., Jones, 2001; Ross, 1987, p. 11   |
| <code>\citeA&lt;e.g.,&gt;[p.~11]{Jone01,Ross87}</code>      | e.g., Jones (2001); Ross (1987, p. 11) |
| <code>\citeauthor&lt;e.g.,&gt;[p.~11]{Jone01,Ross87}</code> | e.g., Jones; Ross, p. 11               |
| <code>\citeyear&lt;e.g.,&gt;[p.~11]{Jone01,Ross87}</code>   | (e.g., 2001; 1987, p. 11)              |
| <code>\citeyearNP&lt;e.g.,&gt;[p.~11]{Jone01,Ross87}</code> | e.g., 2001; 1987, p. 11                |

**bibnewpage**

The bibliography is started on a new page. This is required by some journal styles, including the APA manual. The `apa` class already contained this in its `man` option, but now it has been made available directly in `apacite`.

**nobibnewpage**

The bibliography is not explicitly started on a new page, although if the bibliography is a chapter, it will be started on a new page anyway, because chapters are started on a new page. This is the default in `apacite` and thus is the only time a non-APA setting is used as default instead of an available APA setting. Therefore, to satisfy the APA rules, you have to request the `bibnewpage` option explicitly.

## 4 The citation commands

In this section, the commands are described that can be used to cite a work. Their default behavior will be explained. In section ??, customization of this behavior will be discussed. Extensive examples are given in section ?. Table ?? gives a brief impression of the usage and results of the most common citation commands.

If the `apacite` package is loaded, the following citation commands can be used.

`\cite<prefix>[<postfix>]{<keys>}`

`\fullcite<prefix>[<postfix>]{<keys>}`

`\shortcite<prefix>[<postfix>]{<keys>}`

These three commands produce parenthetical citations of the following form: “(*<prefix>* *<Authors1>*, *<Year1>*; *<Authors2>*, *<Year2>*; ...; *<AuthorsN>*, *<YearN>*, *<postfix>*)”. The command `\fullcite` uses the “full” author list, and the `\shortcite` command uses the abbreviated author list. The `\cite` command uses the “full” list the first time a work is referenced and the “short” list the second and subsequent times. In accordance with the APA manual, the “full” list consists of all authors if their number is five or less, and the first author suffixed with “et al.” if their number is six or more. Similarly, the “short” list consists of all authors if their number is two or less and the first author with “et al.” if their number is three or more. There are some nuances for complicated situations. These will be discussed in section ??.

If the (full) author lists of subsequent citations within the same citation command are the same, they are not repeated; only the years of the later citations are given. This holds for

all analogous cases as well.

The prefix and postfix are optional. Thus, e.g., `\cite{<keys>}`, `\cite[<postfix>]{<keys>}`, and `\cite<prefix>{<keys>}` are also allowed. This holds for all analogous cases as well.

`\citeNP<prefix>[<postfix>]{<keys>}`

`\fullciteNP<prefix>[<postfix>]{<keys>}`

`\shortciteNP<prefix>[<postfix>]{<keys>}`

“No parentheses” versions of `\cite`, `\fullcite`, and `\shortcite`. They produce citations of the form “<prefix> <Authors1>, <Year1>; <Authors2>, <Year2>; ...; <AuthorsN>, <YearN>, <postfix>”. Can be used for complex citations within parenthetical material, see the discussion of `\citeyearNP` below.

`\citeA<prefix>[<postfix>]{<keys>}`

`\fullciteA<prefix>[<postfix>]{<keys>}`

`\shortciteA<prefix>[<postfix>]{<keys>}`

These commands are similar to the commands without the “A” suffix. They produce in-text citations of the form “<prefix> <Authors1> (<Year1>), <Authors2> (<Year2>), ..., <AuthorsN> (<YearN>, <postfix>)”. Typically, the <prefix> will be empty, because it is part of the main text, and there typically will only be one citation in <keys> (unless the authors of multiple citations are the same), because the authors’ names are also part of the main text. But it was easy to provide the additional options, and this enhances the similarity with the other citation commands, which I found desirable.

`\citeauthor<prefix>[<postfix>]{<keys>}`

`\fullciteauthor<prefix>[<postfix>]{<keys>}`

`\shortciteauthor<prefix>[<postfix>]{<keys>}`

`\citeauthorNP<prefix>[<postfix>]{<keys>}`

`\fullciteauthorNP<prefix>[<postfix>]{<keys>}`

`\shortciteauthorNP<prefix>[<postfix>]{<keys>}`

These commands are similar to their `\cite`-type counterparts, except that they produce citations of the form “<prefix> <Authors1>, <Authors2>, ..., <AuthorsN>, <postfix>” (the years and parentheses are left out). This is typically used in a paragraph when a proper citation (incl. year) to the same work has already been given in the paragraph.<sup>1</sup> In such a case, according to the APA manual, the year must not be repeated for repeated references to the same work in the same paragraph.

A second important application of these commands is when some complex citations must be constructed, like “?’s (?) theory”, which makes combined use of `\citeauthor` and `\citeyear`, see below.

Normally, I would expect only one citation per `\citeauthor`-type command, and no <prefix> or <postfix>.

---

<sup>1</sup>Note that a paragraph is not a section. Nor does it need to be declared as a `\paragraph`. A reminder for Dutch readers: “section” = “paragraaf” and “paragraph” = “alinea”.

The difference between the “non-NP” versions and the “NP” versions is that the latter are intended to be used parenthetically, like `\cite`, whereas the former are intended to be used in the running text, like `\citeA`. Neither of these generate the parentheses, though. Thus, the only real difference is that the “non-NP” versions use “and” between authors and the “NP” versions use “&”.

Following the `\cite`–`\citeNP`–`\citeA` logic, `\citeauthor` should actually be called `\citeauthorA` and a `\citeauthor` command should be supplied that is the same as the current `\citeauthorNP` but with parentheses added. For historical reasons and some compatibility with other citation packages, this is currently not the case. I may try to invent a decent solution for a next release of `apacite`.

`\citeyear<⟨prefix⟩>[⟨postfix⟩]{⟨keys⟩}`

Produces citations of the form “(⟨prefix⟩ ⟨Year1⟩, ⟨Year2⟩, ..., ⟨YearN⟩, ⟨postfix⟩)”. See the discussion of `\citeauthor` above. Typically, the ⟨prefix⟩ would be empty and multiple citations would only be used if they have the same authors.

`\citeyearNP<⟨prefix⟩>[⟨postfix⟩]{⟨keys⟩}`

A “no parentheses” version of `\citeyear`. Produces citations of the form “⟨prefix⟩ ⟨Year1⟩, ⟨Year2⟩, ..., ⟨YearN⟩, ⟨postfix⟩”. Can be used for complex citations within parenthetical material, like “the work of ? (?: but see also ?, ?, ?)”, which also uses the `\citeNP` command.

`\nocite{⟨keys⟩}`

The entries are included in the bibliography without being cited in text. This is standard L<sup>A</sup>T<sub>E</sub>X. Note that the APA manual normally does not allow this. The only exception concerns works that are studied in a meta-analysis, but for these the `\nocitemeta` command should be used.

`\nocitemeta{⟨keys⟩}`

This is similar to `\nocite`. It is used to get the list of works included in a meta-analysis in the bibliography. This command has the additional effect that the corresponding items in the bibliography are preceded by an asterisk (\*) and a remark explaining this is inserted just before the bibliography, as required by the APA manual.

In order to format the examples from the APA manual correctly in section ??, the `apacdoc.sty` package defines the following additional citation commands: `\DSMcite`, `\biblecite`, and `\citesong`, and obvious variations with `full`, `short`, `A`, and `NP`. I have decided not to include these in `apacite.sty`, primarily because I find it unattractive to introduce different types of citation commands for different types of citations. This also makes it more difficult to use the same document with another citation package. Perhaps I’ll be able to provide, in a future version of `apacite`, more elegant solutions to the problems for which these commands were defined. See section ?? for a description of the `apacdoc.sty` package and some brief usage instructions.

## 5 Contents of the bibliography database file

The information that is used by L<sup>A</sup>T<sub>E</sub>X/BIB<sub>T</sub>E<sub>X</sub> 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 ? (? , section 12.2) and ? (? , section 13.2). Roughly speaking, the `.bib` file consists of a collection of *entries* of the form

`@⟨entryname⟩{⟨contents⟩}`

Most entries describe a work that may be referenced, e.g., a book or article. There are, however, two exceptions: a **preamble** entry (with  $\langle entryname \rangle = \text{preamble}$ ) and a **string** entry (with  $\langle entryname \rangle = \text{string}$ ). There is typically at most one **preamble** entry. Its  $\langle contents \rangle$  consist of a string—according to  $\text{BIB}_{\text{TEX}}$ , i.e., between double quotes (") or an additional pair of braces ({ and }). This string, which typically consists of  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  commands like `\newcommand{\SortNoop}[1]{}`, is literally included in the document before the bibliography. It can therefore be used to define commands that are used in the (other) entries of the `.bib` file and that are not standard  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .

A **string** entry is similar, but the contents are now not included in the document. Instead, the string is assigned to a kind of “variable”, the name of which may contain characters not usually associated with variable names, like colons and hyphens, see ? (, p. 769). An example of a **string** entry is

```
@string{ JPSP = {Journal of Personality and Social Psychology} }
```

Then, if in a later entry, the journal is *Journal of Personality and Social Psychology*, you can format this as

```
journal = JPSP,
```

instead of the usual

```
journal = {Journal of Personality and Social Psychology},
```

This is not only convenient because it saves typing of common long journal names (and reduces the likelihood of typos), but it can also be used to define style-specific variations. For example, some styles abbreviate journal names, such as *J. Pers. Soc. Psych.* instead of *Journal of Personality and Social Psychology*. You could then put the definitions of all full journal names in one `.bib` file (`fulljou.bib`, say) and the definitions of the abbreviated journal names in another (`abbrjou.bib`, say). Suppose that the information about the referenced works is stored in a third file, say, `myrefs.bib`, and that the “NotAPA” style rules, implemented in the `notapa.bst`  $\text{BIB}_{\text{TEX}}$  style file, require the abbreviated journal names. Then you can use

```
\bibliographystyle{notapa}
\bibliography{abbrjou,myrefs}
```

to obtain a reference list complying with the “NotAPA” rules. If you change your mind and decide to switch to using the rules of the APA manual, you only need to change the two lines above into

```
\bibliographystyle{apacite}
\bibliography{fulljou,myrefs}
```

(and additionally load the `apacite.sty`  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  package through `\usepackage`).

The remaining entry types are types that correspond with the type of a referenced work, e.g., **book** or **article**. These entry types (reference types) are discussed in section ?? below. Such entries have the following structure:

```
@⟨entryname⟩{⟨key⟩,
  ⟨fieldname1⟩ = ⟨value1⟩,
  ⟨fieldname2⟩ = ⟨value2⟩,
  :
  :}
```

```

    <fieldnameN> = <valueN>,
}

```

where  $\langle key \rangle$  corresponds to the key used in the citation commands and matches the entry with the citation (and should therefore be unique). The fieldnames are **author**, **year**, etc., which are described in section ?? below. The values are strings, either defined previously by a **string** entry or explicitly indicated as such here by putting the relevant information between double quotes (") or a pair of braces ({ and }).

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 (e.g., ?, ?, chap. 13). Furthermore, the meaning and usage of many fields and reference types that *are* described there have been altered somewhat. Therefore, a complete description is given here. The symbol <sup>†</sup> will be used to indicate that the field or reference type is also described in ? (? , Appendix B), ? (? , chap. 12), or ? (? , chap. 13), although, as mentioned above, the specific meaning of the item may have been changed.

## 5.1 Types of references

In this section, the list of reference types that are recognized by **apacite** is given. An overview of which fields can be used for which reference type is given in Table ??. If fields are not relevant, but you use them anyway in your **.bib** file, they are ignored. The fields and their use are described in more detail in section ??, but here some specific issues will already be mentioned.

The following entry types (reference types) are defined in **apacite**:

### **article**<sup>†</sup>

A journal article or comparable. If the “article” referenced to is a special issue of a journal or something else that has editors instead of authors, the **author** field should be empty and the **editor** field should be used for the editors. If the journal paginates by issue instead of by volume, or when you are referring to a whole special issue, the issue number should be given in the **number** field. In all other cases (including referring to an article within a special issue), the **number** field must not be used. The **type** field can be used to denote the type of article, for example, “**Letter to the editor**”, or “**Review of the book \emph{Life in the middle}**”.

### **magazine**

A magazine article. Unlike in early versions of **apacite**, this is now equivalent to **article**. However, with an **article**, the **month** and **day** fields should generally not be used, unless there is a compelling reason to use them. For a magazine, the **month**, and **day** (for weeklies) fields are commonly used.

### **newspaper**

A newspaper article. This is similar to **article** and **magazine**, except that the pages are formatted a little different, with “pp. 23–49” instead of just “23–49”.

### **book**<sup>†</sup>

An entire book. The **type** field can be used to denote a specific type of item (so generally not strictly a book), e.g., “**Brochure**”. Similarly, the **number** field can be used if this is considered useful. (It is used in example 24 in the APA manual, see below.) This seems quite rare to me, however. It should certainly *not* be used for ISBN numbers and the like.

Table 2: List of fields that are used by the various reference types (blank = not used; × = used; *c* = used, but only for citations; *a* = only used by **apacann** and **apacannx**).

|                   | Reference type |          |              |            |              |          |               |             |          |          |
|-------------------|----------------|----------|--------------|------------|--------------|----------|---------------|-------------|----------|----------|
|                   | article        |          |              | phdthesis  |              |          |               |             | manual   |          |
| Field             | magazine       | book     | incollection | techreport | intechreport | lecture  | mastersthesis | unpublished | misc     | literal  |
| address           |                | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| annotate          | <i>a</i>       | <i>a</i> | <i>a</i>     | <i>a</i>   | <i>a</i>     | <i>a</i> | <i>a</i>      | <i>a</i>    | <i>a</i> | <i>a</i> |
| annotate          | <i>a</i>       | <i>a</i> | <i>a</i>     | <i>a</i>   | <i>a</i>     | <i>a</i> | <i>a</i>      | <i>a</i>    | <i>a</i> | <i>a</i> |
| author            | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| booktitle         |                |          | ×            |            | ×            |          |               |             |          |          |
| chair             |                |          |              |            |              | ×        |               |             |          |          |
| chapter           |                |          | ×            |            |              |          |               |             |          |          |
| day               | ×              |          | ×            | ×          | ×            | ×        |               | ×           | ×        |          |
| edition           |                | ×        | ×            | ×          | ×            |          | ×             |             | ×        |          |
| editor            | ×              | ×        | ×            | ×          | ×            |          |               | ×           | ×        |          |
| englishtitle      | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| firstkey          | <i>c</i>       | <i>c</i> | <i>c</i>     | <i>c</i>   | <i>c</i>     | <i>c</i> | <i>c</i>      | <i>c</i>    | <i>c</i> | <i>c</i> |
| howpublished      | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| institution       |                |          |              | ×          | ×            |          |               |             |          |          |
| journal           | ×              |          |              |            |              |          | ×             |             |          |          |
| key               | <i>c</i>       | <i>c</i> | <i>c</i>     | <i>c</i>   | <i>c</i>     | <i>c</i> | <i>c</i>      | <i>c</i>    | <i>c</i> | <i>c</i> |
| lastchecked       | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| month             | ×              |          | ×            | ×          | ×            | ×        |               | ×           | ×        |          |
| note              | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| number            | ×              | ×        |              | ×          | ×            |          | ×             | ×           | ×        |          |
| organization      |                |          |              |            |              |          |               | ×           |          |          |
| originaladdress   | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originalbooktitle | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originaledition   | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originaleditor    | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originaljournal   | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originalnumber    | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originalpages     | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originalpublisher | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originalvolume    | ×              | ×        | ×            |            | ×            |          |               |             |          |          |
| originalyear      | ×              | ×        | ×            | <i>c</i>   | ×            | <i>c</i> | ×             | <i>c</i>    | <i>c</i> | <i>c</i> |
| pages             | ×              |          | ×            |            | ×            |          | ×             |             |          |          |
| publisher         |                | ×        | ×            |            |              |          |               |             | ×        |          |
| school            |                |          |              |            |              |          | ×             |             |          |          |
| series            |                |          |              |            |              |          |               |             |          |          |
| symposium         |                |          |              |            |              | ×        |               |             |          |          |
| text              |                |          |              |            |              |          |               |             |          | ×        |
| title             | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| translator        | ×              | ×        | ×            | <i>c</i>   | ×            | <i>c</i> | <i>c</i>      | <i>c</i>    | <i>c</i> | <i>c</i> |
| type              | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| url               | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        |          |
| volume            | ×              | ×        | ×            | ×          | ×            |          | ×             |             | ×        |          |
| year              | ×              | ×        | ×            | ×          | ×            | ×        | ×             | ×           | ×        | <i>c</i> |

#### **incollection<sup>†</sup>**

An article in a (usually edited) book, or other kind of larger work, except a report, for which the **intechreport** entry is used. The **booktitle** field contains the title of the whole collection (book). The **type** field is currently used to denote the type of article, as with the **article** reference type, although in some cases, it may be more naturally to let the **type** field contain the type of the whole work (e.g., CD). Therefore, I may change this somewhat in future versions of **apacite**.

#### **techreport<sup>†</sup>**

A report. This may be a “technical” report such as published by universities, or a report from government organizations or private companies. The **type** field can be used to indicate what kind of report it is, e.g., “College Board Rep.” or “{DHHS} Publication”. If the **type** field is missing, the default type “Tech.\ Rep.\hbox{”} is used. (The \hbox{} ensures that T<sub>E</sub>X does not view the period as the end of a sentence, which would insert too much space.) If no type description should be given, \bibnotype should be used as the contents of the **type** field. The **number** field can be used to give the report number. The organization that published the report should be given in the **institution** field.

#### **intechreport**

An article in a (usually edited) report. This is more or less a combination of **incollection** and **techreport**. It uses the same fields as the former to describe the article itself, except **type**, e.g., **booktitle**, which is in this case the title of the entire report, and the same fields as the latter to describe the report. In particular, the **type**, **number**, and **institution** fields are used for the report.

#### **lecture**

A paper presented at a meeting. According to the APA manual, the year and month should be given, but **apacite** also uses the **day** field when available. The **symposium** field can be used for the name of the symposium and the **chair** field for the chair of the meeting. The **howpublished** field should be used to indicate the occasion at which the paper was presented if the **symposium** field is empty, e.g., “Paper presented at the meeting of the American Professional Society on the Abuse of Children”. It can also be used to give additional information about the symposium if the **symposium** field is not empty.

#### **phdthesis<sup>†</sup>**

A doctoral dissertation. The **school** field is used to denote the university for which the thesis was written. The **type** field can be used for the thesis type, e.g., “PhD thesis”. If it is empty, the default type is used, which is “Unpublished doctoral dissertation” or “Doctoral dissertation”, depending on whether the **journal** field is empty or not.

If the thesis is abstracted in *Dissertation Abstracts International* or similar, then the **journal**, **volume**, **number**, and **pages** fields can be used as with journal articles. The **year** field should denote the year of the “journal”. The **originalyear** field should be used to denote the year of the original thesis, even if it is the same as the year of the abstract, provided that the original thesis is used, and not the abstract. Then, the **school** field should also be used, and optionally the **type** field. If only the abstract is used, it should presumably be treated as a journal article.

#### **mastersthesis<sup>†</sup>**



This is equivalent to the `phdthesis` type, except that the default “unpublished” and “published” types are “Unpublished master’s thesis” and “Master’s thesis”.

#### `unpublished`<sup>†</sup>

For unpublished manuscripts and similarly “obscure” material. The `howpublished` field will typically be used to indicate what kind of work is referred to (e.g., “Unpublished manuscript”). The `organization` field can be used to denote the organization in which the document was produced, e.g., “Johns Hopkins University, Center for Social Organization of Schools”. Note that the larger organization should be given first and the department after that. The address (city and state, etc., see the discussion of the `address` field in section ?? below) of the organization should be given in the `address` field, unless it is already mentioned in the name of the organization. Presumably, if the organization is “University of Groningen”, the city name “Groningen” should not be given in the `address` field, but the country “The Netherlands” should. When both the `organization` and the `address` field are available, `apacite` formats these as “organization, address”. This is different from the usual “address: publisher” form. This behavior cannot explicitly be deduced from the 5th edition, nor from the 4th edition, of the APA manual, but the 3rd edition (?, ?) gives an explicit example (example 53, p. 131): ? (?).

#### `misc`<sup>†</sup>

For works that do not fit into the other categories, such as motion pictures, cassette recordings, computer software, etc. The `type` field can be used to indicate the type of work, and `apacite` recognizes a few specific types that should be formatted a little differently, see the discussion of the `type` field below. If the `address` and `publisher` fields are available, the address and publisher part is formatted as with books. The `unpublished` and `misc` types are very similar, but there are some differences. For example, `unpublished` uses the `organization` field and `misc` uses the `publisher` field, and this part is formatted differently. Further, the `unpublished` type requires `author`, `editor`, `title`, or `type`, whereas with `misc`, `howpublished` takes the first position if these fields are all empty. Finally, as briefly indicated above and discussed in more detail below, `misc` recognizes some special types in the `type` field.

#### `literal`

If the other categories do not format the item correctly, this category can be used. The `text` field is copied literally to the bibliography. The `firstkey`, `key`, `year`, and possibly `originalyear` and `translator` fields are necessary to get correct in-text citations. However, I have never needed to use this entry type and I think that `misc` should also be able to format the entry correctly. Furthermore, the correct placement of `literal` references in the reference list (alphabetizing, sorting), may be problematic.

#### `manual`<sup>†</sup>

This is treated as a `misc`, except that when the `type` field is empty, it is assumed to be a `\bibcomputersoftwaremanual`. See also the description of the `type` field below.

#### `booklet`<sup>†</sup>

#### `inbook`<sup>†</sup>

#### `inproceedings`<sup>†</sup>

### proceedings<sup>†</sup>

These categories are defined by all standard citation styles. They are, however, not needed for the examples in the APA manual. For compatibility, however, they are included and defined as follows: `booklet = proceedings = misc`; `inbook = inproceedings = incollection`. ? (, p. 230) also mention `conference` as being the same as `inproceedings`; for compatibility, this is now supported in `apacite`.

## 5.2 Fields

The following fields can be used to describe a reference in the `.bib` file:

### address<sup>†</sup>

The address (usually the city and state or country) of the publisher, school, institution, or organization that published the item or at which the item was produced. The APA manual requires that the state or territory of a U.S.-city must be given in the official two-letter U.S. Postal Services form. Only for a specific list of 17 specific cities that are “well known for publishing” can (must) the state and/or country description be omitted.

### annotate<sup>†</sup>

### annotate<sup>†</sup>

These are used with the `apacann` and `apacannx` bibliography styles to generate annotated bibliographies, see section ?? . They are simply ignored without warning by the `apacite` and `apacitex` bibliography styles.

### author<sup>†</sup>

The author(s) of the work. This may also be a corporate author when applicable, but some specific measures must then be taken to prevent the corporate name from being interpreted as a first name and last name (such as “Association, A. P.”):

```
author = {{\bibcorporate{American Psychological Association}}},
```

It is important to surround the `\bibcorporate{<Author>}` by an additional pair of braces. Otherwise, `BIBTEX` sorts the entry as if its name was “`Bibcorporate{<Author>}`”. Section ?? contains more examples.

As discussed in the standard `BIBTEX` documentation, multiple authors must be separated by the word `and`, and each author’s name can either be given in the form “`<firstnames> <lastnames>`” or “`<lastnames>, <firstnames>`”. However, “von” parts and “junior” parts complicate issues a bit. See the examples and the discussion of them.

I would generally give the full first name(s) in the `.bib` file, even though the APA manual only requires initials, because other styles require full first name(s) and `apacite` abbreviates to initials automatically. I have done this only occasionally with the examples, because the APA manual only gives the initials. Therefore, I have only provided first names with a few examples for which I knew the authors’ first names.

If there are more than 6 authors, the APA manual requires that the first 6 should be named in the reference list, followed by “et al.”. Therefore, `apacite` uses only the names of the first 6 authors, and inserts an “et al.” when applicable. Therefore, in the `.bib` file, the first 6 (or more) authors may be given (separated by “and”), followed by “and others”. This phrase

is recognized by `apacite`. It is, however, better to give all authors in the `.bib` file, so that each style can select its own truncation number.

**booktitle<sup>†</sup>**

The title of the larger work, typically book, but sometimes something else, like a report, in which the referenced item (article) was published.

**chair**

The chair(s) of a symposium or meeting. Used for lectures. It is formatted the same way as editors.

**chapter<sup>†</sup>**

The chapter number if the referenced item is a chapter in a larger collection. Typically used if the referenced work is a chapter in an internet document. If the collection is a book, page numbers (in the `pages` field) should be used instead according to the APA rules.

**crossref<sup>†</sup>**

The `apacite` package does not recognize this field, because the APA manual does not discuss explicit cross-referencing. If the referenced work is a chapter (article) in an edited book, all relevant information should be given in the reference list as part of the information about the referenced chapter, not as, e.g., “In Wainrib (1992)”. If several chapters from the book edited by Wainrib are referenced, the same information about this book is given with each referenced chapter, and the work itself is not a separate entry in the reference list (unless it is explicitly referred to).

However, the `crossref` field *can* be used, as part of standard `BIBTEX` usage. Missing fields for the referenced work are then copied from the cross-referenced entry, which must come *after* the referring entry in the `.bib` file. If you use this, remember to put the title of the book (also) in the `booktitle` field, because the `title` field of the referring entry is not empty (it contains the chapter title).

I have not tested cross-referencing in detail, so I do not know whether it behaves well (i.e., complies with the implicit or explicit APA rules) under various circumstances. Therefore, I do not recommend using it without thorough testing.

**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.

**edition<sup>†</sup>**

The edition of the book or report. Traditionally, this is of the form “1st”, “2nd”, “3rd”, etc., or “Rev.” for a revised edition. This form is still supported, but `apacite` now also supports the form in which only the number is given: “1”, “2”, etc. If `apacite` detects that only a number is given, it inserts “`\PrintOrdinal{number}`” in the `.bbl` file. This `LATEX` function converts the number again to “1st”, etc. The advantage of this is that the same bibliography database can be used for documents in different languages: In the language-specific (`.apc`) files supplied with `apacite`, `\PrintOrdinal` is redefined according to the document language.

**editor<sup>†</sup>**

The editor(s) of a book, report, or special issue of a journal. In the examples, this field is also “misused” for the producer of a television series.

#### **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. For an article in a journal with a non-English name or in a book with a non-English title, the journal name or book title should not be translated or put in this field, only the title of the referenced work itself must be translated. Because, as of version [2005/06/01], **apacite** contains some support of other languages for the main document (see section ??), this field needs rethinking, which will be deferred to a future version (but, as always, suggestions are welcome).

#### **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. See also **key**.

#### **howpublished<sup>†</sup>**

A description of how something was “published”. For example, “Unpublished manuscript” or “Paper presented at the meeting of the American Professional Society on the Abuse of Children”. Also used for more retrieval information about electronic documents, e.g., “Retrieved November 19, 2000, from the PsycARTICLES database”, or

```
howpublished = {Retrieved August 24, 2000, from Columbia University,  
                Institute for Learning Technologies Web site:  
                \url{http://www.ilt.columbia.edu/  
                publications/papers/newwine1.html}\bibnodot{.}}
```

However, if the retrieval information consists only of a date and/or a URL, it is better to use the **lastchecked** and **url** fields for this and let **apacite** format this information. See also the **note** field for an alternative to the **howpublished** field.

#### **institution<sup>†</sup>**

The institution, university, or company that published a (technical) report.

#### **journal<sup>†</sup>**

The journal, magazine, newspaper, etc. in which an article, a review, or an abstract of a thesis was published.

#### **key<sup>†</sup>**

See **firstkey**. The **key** field, if not empty, is used for second and subsequent citations, or all citations if **firstkey** is missing. If **author** is empty and **editor** is empty or can not be used as author (e.g., in **incollection**), the **key** field may be necessary to obtain a useful citation. This is so, because the APA manual requires an abbreviation of the title as in-text citation, and the abbreviation must be sensible. In the current version, **apacite** uses the whole title if **firstkey** and **key** are missing. Therefore, if the title is considered too long to use in citations, the user must provide an abbreviation in the **key** field. Furthermore, the user must then also define the formatting: If the key is used for a difficult kind of author (e.g., corporate author), it should be formatted as an author, i.e., in plain text, with names

capitalized. If the key is used for an abbreviation of an article title, it should be in plain text, with major words capitalized (unlike in the reference list), and between double quotes (“ and ”; also unlike in the reference list). If the key is used for an abbreviation of a book title, it should be emphasized, with major words capitalized (unlike in the reference list), but not between quotes.

In a field in the `.bib` file that is also used for the citations, such as the `key` field, formatting through, e.g.,

```
key = {\itshape Text},
key = {\em Text\},
```

can be done (note the extra pair of braces!), but the seemingly more logical variations

```
key = {\textit{Text}},
key = {\textit{Text}},
key = {\emph{Text}},
key = {\emph{Text}},
```

do not work. However, it is generally preferable to use as less explicit formatting in the `.bib` files as possible, because this decreases the possibilities of successfully using the same `.bib` file with other styles. The following works excellent:

```
key = {\APACcitebtitle{Text}},
```

where the `\APACcitebtitle` macro recognizes that the argument should be formatted as a booktitle (i.e., in italics according to the APA rules). This macro, and its companion `\APACciteatitle` for article title formatting, are used by `apacite` if the `firstkey` and `key` fields are missing, but can also be used by the user. Note that if you use such a `.bib` file with another style, you have to provide definitions of these macros yourself (which can be copied from the `apacite` source code).

*Warning:* the `key` field, which acts as a kind of pseudo-author, should not be confused with the (citation)  $\langle key \rangle$  that is used to match citations with entries in a `.bib` file.

#### **lastchecked**

The date on which a web page was last visited. Only used in combination with the `url` field. Currently, `apacite` does not contain language-specific code to dissect this date, but copies its contents literally to the `.bbl` file. Hence, if you intend to use the same bibliography database for documents in different languages, you may have to use a self-defined command for this. You can use the `apacite` commands `\APACrefYearMonthDay` and `\APACmonth` as templates for this. I will probably provide such a facility in a future version of `apacite`.

#### **month<sup>†</sup>**

The month something was published. Mainly used for magazine or newspaper articles, lectures, and electronic documents. Use the month macros `jan`, `feb`, etc. These are predefined strings in the `.bst` files, reverting to “`\APACmonth{01}`”, etc. You can also use the additional macros `winter`, `spring`, `summer`, and `fall`, which the `.bst` files translate into “`\APACmonth{13}`”—“`\APACmonth{16}`”. The `\APACmonth` command translates the numbers to the month or season names. This system allows language-specific behavior at the L<sup>A</sup>T<sub>E</sub>X stage.

#### `note`<sup>†</sup>

A note. This puts additional information between parentheses at the end of a reference list entry. In the examples from the APA manual, it is used for NTIS No., ERIC No., and UMI No., and for notes about how to obtain the work (“Available from *⟨organization⟩*, *⟨full address⟩*”). If however, the work has been obtained from the internet or from an “aggregated database”, then the `howpublished` or `url` field should be used, because it should not be between parentheses then.

Finally, the `note` field is used to give the recording date (i.e., year) of a song when this is different from the date (year) of copyright. The latter should be put in the `year` field.

Do not use the `note` field to denote the original publication of a reprint or translation, because the `originalyear` field must be used in those cases to obtain the correct citation in the text (e.g., ?, ?).

#### `number`<sup>†</sup>

The number of a journal issue or a report. This can be quite complex, for example, “PRM~92-01”, or “3, Pt.~2”, or “1, Serial No.~231”. These latter examples imply that the generality (language and style independence) of the `.bib` file is somewhat lost. Furthermore, if the journal paginates by year and not by issue, the issue number should not be mentioned at all (except when referring to a whole special issue). Of course, `apacite` does not know whether the journal paginates by issue or by year. If the `number` field is available, `apacite` simply assumes that it should be included. It is the user’s responsibility to ensure that this is indeed the case.

#### `organization`<sup>†</sup>

Used for the `unpublished` reference type to denote the organization in which the unpublished work was produced.

`originaladdress`

`originalbooktitle`

`originaledition`

`originaleditor`

`originaljournal`

`originalnumber`

`originalpages`

`originalpublisher`

`originalvolume`

`originalyear`

These (`original*-`) 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. This can be used in referring to translations or reprints of articles in journals or books. If the original work is a book or other “standalone” work, only the `originalyear` field should be used. If the original work is a (PhD or Master’s) thesis that is also abstracted

in *Dissertation Abstracts International*, *Masters Abstracts International*, or comparable, the **originalyear** field should be used to denote the year of the dissertation and the **year** field to denote the year of publication of the abstract, even if they are the same.

**pages<sup>†</sup>**

The (inclusive) page numbers of the article that is referred to, in the journal or book in which it was published. If a range of pages is given (which is usually the case), an en-dash should be used: “29--43”. Other styles sometimes allow a single hyphen (“29-43”), which is then automatically formatted as an en-dash (“29–43”), rather than as a hyphen (“29-43”), but I have encountered situations, such as software manuals (LISREL) and reports for the U.S. Government, in which the page number was of the form “II-3”, meaning page 3 of chapter 2. In such a situation, you may get page numbers like “II-1-II-15”, where the distinction between a hyphen and an en-dash becomes important. Therefore, **apacite** does not perform such automatic transformations.

**publisher<sup>†</sup>**

The publisher of the item. Primarily used for books.

**school<sup>†</sup>**

The school or university for which a PhD thesis or master’s thesis was written.

**series<sup>†</sup>**

Not used, although the APA manual has a rule for series. If you refer to a volume in a series, the series title, volume number, and volume title should be joined into a two-part title, e.g., *Handbook of child psychology: Vol. 4. Socialization, personality, and social development*, see ex. 36. In **apacite** this whole part must be put in the **title** or **booktitle** field (whichever is appropriate).

For a long time, I have intended to support the **series** field to allow disentangling such situations. However, in the end, I have decided not to do this. The reason for this is that, according to my interpretation, the meaning of “series” in the APA manual is quite different from the meaning of the **series** field, as intended by Oren Patashnik (the author of **BIBTEX** and the original style files). Many books are published in some sort of series. For example, my book with Tom Wansbeek (?), is vol. 37 in the *Advanced Textbooks in Economics* series, of which the series editors are C. J. Bliss and M. D. Intriligator. The intention of the **series** field is to generate a reference list entry like

Wansbeek, T., & Meijer, E. (2000). *Measurement error and latent variables in econometrics* (Advanced Textbooks in Economics No. 37). Amsterdam: North-Holland.

In economics (and often in psychology, I guess), mentioning the series usually does not add much relevant information, and it is therefore typically omitted. I think this is how the APA intends it as well. However, in computer science, it is often very convenient, because certain series, like Springer’s *Lecture Notes in Computer Science* (LNCS) have a certain status of their own, and researchers know how to find volumes in this series.

In my interpretation, the APA “series” refers to a single multivolume work with separate editors for the constituent volumes, but with a series editor who initiates the whole series. In the example from the APA manual, the *Handbook of child psychology* is not a loose collection of separate books by different authors, but a combined effort managed by the series editor to give an overview of the field. In this case, the series editor may be considered the primary

author and it makes sense to have the different volumes of the series together in the reference list. With my book with Tom Wansbeek, the series editors act more like journal editors, and different volumes of the series are unrelated books that do not need to be gathered together in the reference list. It would make no sense to cite this book as “Bliss, Intriligator, Wansbeek, & Meijer (2000)”.

Because in my interpretation, the usual meaning of the `series` field should not be included in the reference list according to the APA rules, `apacite` ignores the `series` field. Perhaps I'll implement the APA manual “series” rules by introducing a `volumetitle` field, but this will further reduce compatibility with other citation styles.

#### `symposium`

The name of the symposium or meeting at which a lecture was given. This is typically used for the construction “In *⟨chair⟩* (Chair), *⟨symposium⟩*”, see ex. 51. I have never used it myself, however, I always use the “Paper presented ...” form, which puts this information in the `howpublished` field.

#### `text`

Used for items of type `literal`. This field contains the complete literal text to be used in the bibliography. I've never used it.

#### `title`<sup>†</sup>

The title of the work.

#### `translator`

The translator of a book or article. This should be formatted in the same way as `author` and `editor`. If the `translator` and `editor` fields are identical, they are formatted as if there were only an editor, except that the editor receives the suffix “(Ed. & Trans.)” instead of just “(Ed.)”, or similarly if there is more than 1 editor.

#### `type`<sup>†</sup>

The type of `phdthesis` (e.g., “Doctoral dissertation”), type of `article` (e.g., “Letter to the editor”), type of `techreport` (e.g., “DHHS Publication”), type of `book` (e.g., “Brochure”), type of `misc` (e.g., “Cassette recording”), and so forth.

With a review (which is typically an `article`), the `type` field contains a lot of information and formatting, e.g.:

```
type = {Review of the book {\APACcitebtitle{Life in the middle:
    Psychological and social development in middle age}}},
```

where I used the `\APACcitebtitle` command introduced above in the discussion of the `key` field.

In one of the APA manual examples, I used the `type` field to give a description of the subject of an unpublished raw data file (“Auditory response latencies in rat auditory cortex”), leaving the `title` field blank.

For most reference types, if the `type` field is missing, it is simply ignored and no type description is given. For a `techreport`, however, if the `type` field is missing, the default type (“Tech. Rep.”) is inserted. In ex. 42 of the APA manual, a report without a type description



is given. To make it possible to format a report successfully without a type description, `apacite` recognizes

```
type = {\bibnotype},
```

indicating that there should be no type description.

Another special purpose use of the `type` field is for a computer program, software, programming language and/or manual. These are put into a `misc` entry, but unlike other `misc` types (such as motion pictures) the titles of these types should not be italicized. This is now recognized by `apacite` in the following way: When referring to a computer program, software, programming language and/or manual, you should use the `misc` reference type with one of the following commands in the `type` field:

```
\bibcomputerprogram,  
\bibcomputerprogrammanual,  
\bibcomputerprogramandmanual,  
\bibcomputersoftware,  
\bibcomputersoftwaremanual,  
\bibcomputersoftwareandmanual, or  
\bibprogramminglanguage.
```

Using any of these commands as type designator of a `misc` entry ensures that the title is not italicized, as required. The `manual` entry is now equal to `misc` with default type `\bibcomputersoftwaremanual` if the `type` field is empty.

Similarly, if you refer to a message that has been posted to a newsgroup, internet forum, etc., you should use the `misc` reference type with `\bibmessage` as `type` field.

#### `url`

The web address of a document. Used if the cited document (or other work) is a website or was obtained from a website. Long web addresses can cause problems, because `BIBTEX` splits long lines (more than 80 characters) in the `.bbl` file and inserts a `TEX` comment (percent sign). In most situations, this works fine, because the percent sign “hides” the line break, which would be processed as an undesired space by `TEX`. But within the argument of a `\url` command, a percent sign is copied to the output, because it is a valid character in a URL. To prevent this problem, insert line breaks and/or spaces in the URL in the `.bib` file:

```
url = {ftp://ftp.princeton.edu/harnad/Psycoloquy/  
      2000.volume.11/psyc.00.11.001.  
      language-sex-chromosomes.1.crow},
```

Then `BIBTEX` breaks lines at the spaces/line breaks if a line break in the output is considered necessary (but not otherwise). These spaces are suppressed by default in the output, which is usually the desired behavior. But see the `hyperref` and `url` documentation if you want this differently. Also note that in the `url` field, special characters like ‘%’ and ‘#’ should be entered without a backslash, unlike the normal situation if they would be used in the text.

#### `volume`<sup>†</sup>

The volume or volumes of the referenced book(s) or of the book in which the referenced article was published, or the volume of the journal (magazine, newspaper, ...) in which the referenced article was published.

year<sup>†</sup>

The year in which the referenced item was published, or if it was not published, the year in which it was written or presented. For manuscripts or books that are accepted for publication but have not yet been published, “in press” should be used according to the APA rules. The best way to do this is to use the “\BIP” command, which can be redefined by the user if the language or editorial style requires something else than “in press”. Furthermore, \BIP is recognized by `apacite` and treated a little differently, because for multiple “in press” references with the same author(s), a hyphen (“-”) should be inserted between the “year” (i.e., “in press”) and the “a”, “b”, etc., that follow it to distinguish the works, whereas this hyphen must be omitted if the “year” is an ordinary year (1991a, 1991b).

If no date is explicitly given in (or on) the referenced work, a “n.d.” (no date) should be given as year description. For this, `apacite` supports the `\bibnodate` command. This sometimes leads to different formatting as well, e.g., by referencing a translation of a work of which there is no original date. Then the `originalyear` field should be `\bibnodate` and the `year` field is then, e.g., 1931. If the `translator` field is not empty, the citation in the text is then formatted as, e.g., “? (?)”.

### 5.3 Overriding the default sorting orders

Occasionally, it may be necessary to override BibTeX’s default sorting. For example, the way BibTeX handles accents and other nonstandard characters, which is typically fine for documents in English, may not be acceptable for documents in other languages: “\{u}” is treated as “u”, “\ss” is treated as “ss”, and “\ae” is treated as “ae”. In non-English languages this sorting may not be appropriate and one may 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.

An issue that occurs in English (APA) documents is that, according to the APA, numbers should be sorted as if they were spelled in words: “2003” must be sorted as if it says “twothousandandthree”; see example 4.04 in section ?? below.

Overriding the default sorting order can be done by using the `\APACSortNoop` command in the `.bib` file. This is a L<sup>A</sup>T<sub>E</sub>X 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<sub>llg</sub>”, whereas

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

is sorted as “muellermuller<sub>llg</sub>”, 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<sub>llg</sub>”, which is incorrect.

## 6 Customization

The description in the previous sections, with the exception of section ??, primarily discussed the default behavior of `apacite`. However, as mentioned in the introduction, in addition to the options, `apacite` offers many possibilities for customization. Most punctuation used in the citations and reference list are implemented through L<sup>A</sup>T<sub>E</sub>X 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. Analogously, most fixed texts, like “Tech. Rep.” and “Eds.” are implemented through L<sup>A</sup>T<sub>E</sub>X commands as well, and can similarly be changed by the user. The commands used by `apacite` are discussed in this section. Of course, the defaults are based on the rules of the APA manual.

Some commands, predominantly punctuation, are used both in citations and in the reference list. If you want their definitions in citations to be different from their definitions in the reference list, you can simply redefine their definitions before starting the bibliography.

### 6.1 Punctuation and small formatting issues

The following punctuation commands are provided and used for the citations and reference list.

**\BAsyle** This defines the text style of the authors (or whatever takes their place) for an in-text citation. It defaults to nothing: The authors use the same fonts as the text surrounding it. Introduced because some journals use a different style. For example, *Statistica Neerlandica* uses small-caps, so for that journal, you would define

```
\renewcommand{\BAsyle}{\scshape}
```

Do not use the `\textsc`-type commands, but `\scshape`, `\bfseries`, etc. The `\BAsyle` command is used for the author-part by the citation commands `\cite`, `\shortcite`, `\fullcite`, `\citeA`, `\shortciteA`, `\fullciteA`, `\citeNP`, `\shortciteNP`, and `\fullciteNP`.

**\BAasyle** This is the same as `\BAsyle`, except that `\BAasyle` is used for `\citeauthor`, `\shortciteauthor`, and `\fullciteauthor`.

**\BBOP** Open parenthesis, used for parentheses opening a citation, as in “(Rao, 1973)” or “Rao (1973)”, and the year in the reference list, as in “Rao, C. R. (1973).” Default is “(”.

**\BBCP** The corresponding closing parenthesis. Default is “)”.

**\BAP** This command is inserted after the prefix and before the first citation in a `\cite` command. It defaults to an ordinary space.

**\BBAA** Last “and” between authors in a citation between parentheses and in the reference list, as in “(Mooijaart & Bentler, 1986)” or “Mooijaart, A., & Bentler, P. M. (1986).” Default is “&”.

**\BBAB** Last “and” between authors in a citation in text, as in “Mooijaart and Bentler (1986).” Default is “and”.

**\BBAY** Punctuation between author(s) and year in a citation between parentheses, as in “(Rao, 1973)”. Default is “, ”.

- `\BBYY` Punctuation between two subsequent years, if two works by the same author(s) are referenced in a single citation command, as in “(Rao, 1965, 1973)” or “Rao (1965, 1973)”. Default is “, ”.
- `\BBC` Punctuation between multiple cites, as in “(Rao, 1973; Mooijart & Bentler, 1986)”. Default is “; ”.
- `\BBN` Punctuation before a note (postfix) after a citation, as in “(Rao, 1973, chap. 2)”. Default is “, ”.
- `\BBOQ` Opening quote for an article title in the reference list, as in “<<Random polynomial factor analysis.>>” Default is the empty string: no quotes used.
- `\BBCQ` Closing quote for an article title in the reference list. Default is the empty string. (The `theapa` package required the closing period to be part of the closing quote, but this has been changed, because the period must be left out if title comments, such as type, English translation of non-English title, or translator follow the title.)
- `\BPBI` Period between initials. Default is “.~”, i.e., a period and a non-breaking space, as required by the APA manual. However, many users prefer removing or reducing the space and in some non-English speaking countries this is customary. This can be achieved by redefining this command to, e.g., “.” (no space) or “.\,” (small space). This command cannot be used to remove periods altogether, because there is still a period at the end. I.e., redefining it as an empty string gives awkward results like “Bentler, PM., & Mooijart, A.”.
- `\BHBI` Hyphen between initials. Introduced for analogy with `\BPBI` and used for authors who have two or more first names with a hyphen between them, e.g., Jean-Paul Sartre. Default is “.-”, i.e., a period and a hyphen. Unlike `\BPBI`, I think this one will not be changed often, although `BibTeX` suppresses hyphens when it suppresses spaces between initials as well, so there are people who have different preferences. (Hence, `apacite` partly bypasses the standard `BibTeX` formatting functions to make this work.)
- `\BCBT` Comma between authors in the reference section when the number of authors is two, as in “Mooijart, A., & Bentler, P. M. (1986).” The comma after the “A.” is this one. Default is “,”. The APA manual requires it, but other styles leave out this comma. In such a case, you would redefine this “comma” to be the empty string:
- ```
\renewcommand{\BCBT}{}
```
- `\BCBL` Comma before the last author (for 3 or more authors) in a citation and in the reference section, as in “(Gill, Murray, & Wright, 1981)” or “Gill, P. E., Murray, W., & Wright, M. H. (1981).” Default is “,”. The APA manual requires it, which is standard U.S. usage, but other styles, particularly European, such as British English, leave out this comma. Again, in such a case, you would redefine this “comma” to be the empty string.
- `\BAnd` This is the “and” that is used in the reference list if someone is both editor and translator: “In J. Strachey (Ed. & Trans).” Default is “&”.
- `\theBibCnt` If there are multiple citations with the same author and year, a letter should be added to the year to distinguish the references. For example, one may refer to two or more articles by J. Smith published in 1982. They should be referred to as “Smith (1982a)”, “Smith

(1982b)”, and so forth. To accomplish this, the counter `BibCnt` is defined in `apacite.sty`. The `\theBibCnt` command defines how the value of `BibCnt` is formatted. The default is “`\alph{BibCnt}`”. To emphasize (italicize) the “a”, “b”, etc., you can redefine this as

```
\renewcommand{\theBibCnt}{\em\alph{BibCnt}\/}
```

(This presumably works better than “`\emph{\alph{BibCnt}}`”). To control this behavior, redefining the `\theBibCnt` command should normally be sufficient. However, the complete formatting commands are `\BCnt`, `\BCntIP`, and `\BCntND`. The second of these is for “in press” works, which need an extra hyphen between “in press” and the “a” and “b” suffixes. The third is similar, for works without a date. Their default definitions in `apacite.sty` are

```
\DeclareRobustCommand{\BCnt}[1]{\setcounter{BibCnt}{#1}\theBibCnt}
\DeclareRobustCommand{\BCntIP}[1]{\setcounter{BibCnt}{#1}\mbox{-\theBibCnt}}
\DeclareRobustCommand{\BCntND}[1]{\setcounter{BibCnt}{#1}\mbox{-\theBibCnt}}
```

**\APACciteatitle** The formatting of the title of an article (or similar work) when used as a citation in the text when no author or editor is available for that purpose. Its default definition in `apacite.sty` is

```
\newcommand{\APACciteatitle}[1]{‘‘#1’’}
```

i.e., the title is put between double quotes.

**\APACcitebtitle** The formatting of the title of a book (or other independent work) when used as a citation in the text when no author or editor is available for that purpose. Its default definition in `apacite.sty` is

```
\newcommand{\APACcitebtitle}[1]{\em #1/}
```

i.e., the title is emphasized (in italics) but not put between double quotes.

**\APACmetastar** The asterisk that precedes an item in the bibliography to denote that it is included in the meta-analysis. The default value is “`\ensuremath{{}^{\ast}}`”, which prints as “\*”.

**\bibnewpage** If the `bibnewpage` option is chosen, this command is included before the bibliography. Its default definition is `\clearpage`, but `apa.cls` uses a similar construction with its `man` option through `\newpage`. See ? (, p. 215) or ? (, p. 37) for the differences between `\clearpage` and `\newpage`.

**\bibliographysize** This command is used before the reference list, but after the section or chapter heading. It is intended for the font size of the reference list: For ? (, I defined it as `\small` (and even then the reference list took up 34 pages). The default value is `\normalsize`.

**\bibleftmargin** This gives the indentation of the second and subsequent lines of a reference list entry, relative to the usual left margin. It is not a proper command, but a “skip” (rubber length), which means that it cannot be changed by `\renewcommand`, but by `\setlength`. Its default value is 2.5em.

`\bibindent` This gives the indentation of the first line of a reference list entry, relative to the second line. It is also a “skip”. Its default value is `-\bibleftmargin`, which means that the first line starts at the original left margin, and the second and subsequent lines are indented by 2.5em.

`\bibitemsep` This gives the vertical separation between two reference list entries. It is also a “skip”, with default value `“\z@ \@plus .3\p@\relax”`, i.e., 0pt, but it may be stretched a little to fill the page nicely. This is an old (Plain T<sub>E</sub>X) style definition, I will probably change that in a next version. If you want to change it, e.g., to get a blank line between reference list entries, you can use

```
\setlength{\bibitemsep}{\baselineskip}
```

`\bibparsep` This gives the vertical separation between two paragraphs in the same reference list entry. Such a situation does not occur in the default setup, so it is not used, but it is conceivable that certain customizations lead to its use. It is a “skip” with default value 0pt.

## 6.2 Labels

There are many specific pieces of text that can be put into the reference list or a citation by `apacite`. Here, these pieces are called *labels*. Almost all of these are to some extent language-specific, and sometimes style-specific even within the same language. Therefore, they are implemented through L<sup>A</sup>T<sub>E</sub>X commands, so that users can easily change them. In this section, these commands and their purposes are described, and their (U.S. English) defaults are given. Section ?? below will discuss how these commands are changed if another language is used.

The following label commands are used:

`\bibmessage` This is the first of 9 specific type commands that are recognized by `apacite`. If you define the `type` field of a `misc` entry as `“\bibmessage”`, the formatting of the entry is changed (e.g., the title is not italicized), see the discussion of the `type` field in section ?? above. This one is used for messages in newsgroups, internet forums, etc. Default is `“Msg”`.

`\bibcomputerprogram` Default is `“Computer program”`.

`\bibcomputerprogrammanual` Default is `“Computer program manual”`.

`\bibcomputerprogramandmanual` Default is `“Computer program and manual”`.

`\bibcomputersoftware` Default is `“Computer software”`.

`\bibcomputersoftwaremanual` Default is `“Computer software manual”`.

`\bibcomputersoftwareandmanual` Default is `“Computer software and manual”`.

`\bibprogramminglanguage` Default is `“Programming language”`.

`\bibnotype` This one is a bit different from the previous 8. It is used for `techreport` entries to indicate that the type specifier should be suppressed. Its default value is the empty string, but this will not have an effect on `apacite` behavior, because the type specifier is suppressed anyway by `apacite`. However, by defining it as the empty string, formatting may become better (i.e., closer to the desired formatting) with other styles.

**\bibnodate** Used in the **year** and **originalyear** fields to indicate that no publication date has been given. Default value is “**n.d.\hbox{}**”. The “**\hbox{}**” ensures that **T<sub>E</sub>X** does not treat the last period as a sentence-ending period, after which more space is inserted. This also applies to several other labels discussed below.

**\BOthers** Used for “others” if the number of authors or editors is too large, as in “(Gill et al., 1981)”. The default is “**et al.\hbox{}**”.

However, the definition contains a slight adaptation, because in a previous version of **apacite**, there was a problem implying that in some cases **BIB<sub>T<sub>E</sub>X</sub>** might put an extra period after **\BOthers**, not recognizing that it already contains a period, so that you would get “et al.”, which is undesirable. It depends on the definition of **\BOthers** whether there should be a period or not: If it is defined as “**et al.\hbox{}**”, there should not be an additional period, but if it is defined as, say, “**and others**”, then there should be an additional period. But **BIB<sub>T<sub>E</sub>X</sub>** does not know what the (later) definition of this command will be in **L<sup>A</sup>T<sub>E</sub>X**. Therefore, the **\BOthers** command is defined to have one argument. In **BIB<sub>T<sub>E</sub>X</sub>**, **apacite** inserts **\BOthers{.}** in the output, so no additional period is inserted by **BIB<sub>T<sub>E</sub>X</sub>**. By default, this command is defined as “**et al.\hbox{}**”, and the argument is simply ignored.

If you redefine the **\BOthers** command and need the additional period, you can redefine it appropriately. However, in the current version, the **\BOthers** command is distinguished from the **\BOthersPeriod** command and I think the mentioned problems do not occur anymore. Nevertheless, I have kept the definition of the previous version. I may decide, after thorough testing, to change this again in a next version.

There is still a complication, however. If you use a **\citeauthor** at the end of a sentence, there may be a period too many, if you end the sentence explicitly with it and **\BOthers** is “et al.”. On the other hand, if you omit the sentence-finishing period but decide to redefine **\BOthers** to “and others”, the period is missing. Therefore, it seems wise not to end a sentence with a **\citeauthor**, or you could define a command, e.g., **\finishsentence**, and insert it in an appropriate place. Then, this command should be changed along with **\BOthers**.

**\BOthersPeriod** Also used for “others” if the number of authors or editors is too large, as in “Gill, P. E., et al. (1981)”. But this one is used for situations when it should always end with a period. The default is “**et al.\hbox{}**”.

**\bibcorporate** This command can be used in the **author** field (and **editor** and **translator**) to tell **apacite** that the author is a corporate author. See section ?? for an example of its usage. In combination with the **includecorporate** and **suppresscorporate** options, it can be used to include or exclude corporate authors from the author index. The usage as described above is structured as if **\bibcorporate** has an argument, the corporate author name, but by default, this argument simply would have to be copied to the output without further processing. Therefore, in the actual implementation, **\bibcorporate** does not have an argument and reduces to **\relax**, i.e., it does nothing in **L<sup>A</sup>T<sub>E</sub>X** and exerts its influence at the **BIB<sub>T<sub>E</sub>X</sub>** stage.

**\BIP** “In press”, the string to be used as year for in-press references. In the **.bib** file, the **year** field should be “**\BIP**”, so that **apacite** can recognize this and use appropriate formatting and sorting. The default value is “**in press**”.

**\BIn** Used for `incollection` and `intechreport`, for “In *<editor>* (Ed.), *<booktitle>*”, and similar phrases. Default value is “In”.

**\BCHAP** Used for `incollection` if the `pages` field is empty, to denote the chapter number of the referenced work in the collection. Used primarily for internet documents, where there are no page numbers. Default value is “`chap.\hbox{}`”.

**\BCHAPS** Just like `\BCHAP`, but this one is used if the `chapter` field refers to more than 1 chapter. Default value is “`chap.\hbox{}`”.

**\BED** Editor in reference list, as in “In P. R. Krishnaiah (Ed.)”. Default is “`Ed.\hbox{}`”.

**\BEDS** Editors (more than one editor) in reference list, as in “In C. R. Rao & J. K. Ghosh (Eds.)”. Default is “`Eds.\hbox{}`”.

**\BTRANS** Translator, as in “(J. Strachey, Trans.)”. Default is “`Trans.\hbox{}`”.

**\BTRANSS** Translators. Default is “`Trans.\hbox{}`”.

**\BTRANSL** Translation. Used in a text citation when referring to a translated work for which there is no date for the original work. This is then formatted as “Aristotle (trans. 1931)”. Default is “`trans.\hbox{}`”.

**\BCHAIR** Chair of a symposium, etc. Default is “Chair”.

**\BCHAIRS** Chairs. Default is “Chairs”.

**\BVOL** Volume, as in “Vol. 1”. Default is “`Vol.\hbox{}`”.

**\BVOLS** Volumes, as in “Vols. 1–4”. Default is “`Vols.\hbox{}`”.

**\BNUM** Number, as in “Tech. Rep. No. 12”. Default is “`No.\hbox{}`”.

**\BNUMS** Numbers, as in “Nos. 3–5”. Default is “`Nos.\hbox{}`”.

**\BEd** Edition, as in “2nd ed.”. Default is “`ed.\hbox{}`”.

**\BPG** Page, default is “`p.\hbox{}`”.

**\BPGS** Pages, default is “`pp.\hbox{}`”.

**\BTR** The (default) technical report type name, which can be overridden by the `type` field. Default is “`Tech.\ Rep.\hbox{}`”.

**\BPhD** The (default) PhD thesis type name, which can be overridden by the `type` field. Default is “`Doctoral dissertation`”.

**\BUPhD** The (default) Unpublished PhD thesis type name, which can be overridden by the `type` field. Default is “`Unpublished doctoral dissertation`”.

**\BMTh** The (default) master’s thesis type name, which can be overridden by the `type` field. Default is “`Master’s thesis`”.

**\BUMTh** The (default) unpublished master’s thesis type name, which can be overridden by the `type` field. Default is “`Unpublished master’s thesis`”.



`\BOWP` “Original work published”, default is “Original work published”.

`\BREPR` “Reprinted from”, default is “Reprinted from”.

`\Bby` “by” in the construction “by P. R. Krishnaiah, Ed.” referring to the original editor of a reprinted work. Default is “by”.

`\BAuthor` “Author” as a description of the publisher if the author and publisher are the same. Default is “Author”

`\BAvailFrom` “Available from”, used if the `url` field is specified but the `lastchecked` field is not. Default is “Available from\ ”. Note the extra space. This construction allows customization by redefining this command as being empty, thereby omitting the phrase altogether, without undesirable extra space.

`\BRetrieved` “Retrieved from”, used if both the `url` field and the `lastchecked` field are specified. It has one argument: the date, i.e., the contents of the `lastchecked` field. The default is “Retrieved {#1}, from\ ”, again with extra space.

`\BMsgPostedTo` Used for messages to newsgroups and the like. Default is “Message posted to\ ”, with space.

`\refname` Name of the reference list if it’s a section. (So it’s the section name.) Default is “References”. A problem with the current implementation is, that if you use `babel`’s `\selectlanguage` command, then `\refname` and `\bibname` are overwritten. Because `babel`’s definitions of these commands are often (for various languages) different from the names prescribed by the APA manual and analogous bodies in other countries, it will often be necessary to redefine `\refname` or `\bibname` if you have used `\selectlanguage`. I hope to provide a fix in a next version of `apacite`.

`\bibname` Name of the reference list if it’s a chapter. Default is “References”. See also `\refname`.

`\bibliographyprenote` This command is used before the reference list, but after the section or chapter heading, and immediately after `\bibliographystyle`. It is intended for an optional note just before the references in the reference list. It defaults to nothing, but can be changed to a list of commands (e.g., if some commands should be different in the reference list than in the text) and/or some text.

`\APACmetaprenote` If a meta-analysis is reported, the works included in the meta-analysis should in the bibliography be preceded by an asterisk, see the description of `\APACmetastar` above. This should then be explained in a note before the references. This command contains the text of that note. Default is “References marked with an asterisk indicate studies included in the meta-analysis.”.

`\authorindexname` Name (heading) of the author index. Default is “Author Index”.

### 6.3 More drastic formatting changes to the reference list

In earlier versions of `apacite`, most of the formatting of the reference list was done in the `BIBTEX` stage, i.e., by the `.bst` files. This corresponds with most other `BIBTEX` styles. This makes it difficult for users to customize the formatting of the reference list: They would have to make their

own `.bst` file, by adapting `apacite.bst`. Certain small changes are relatively easy to make, but BibTeX’s “bst” language is not easy to understand for most users, and thus most desired changes, though seemingly small, may be too complicated to implement. Another drawback of this setup is that, each time I release a new version of `apacite`, users would have to make the same adaptations again. In the early days of `apacite`, wasting disk space was another drawback, but this is not important anymore.

To overcome the most salient of these problems, `apacite` has always deferred some formatting choices to the L<sup>A</sup>T<sub>E</sub>X stage, by using tailor-made formatting commands instead of explicit formatting, as discussed in sections ?? and ?? above. The amount of formatting thus handled has slightly grown over time, but still most choices were hard wired in the BibTeX styles.

Since the [2007/09/03] version of `apacite`, a big step has been made in transferring these choices to the L<sup>A</sup>T<sub>E</sub>X domain. Hence, a much larger set of commands, with wider-ranging scope, has been defined, and considerably fewer choices are made at the BibTeX stage. For example, `apacite.bst` now inserts

```
\APACjournalVolNumPages{\journal}\{volume}\{number}\{pages}
```

in the `.bbl` file for an `article` entry, and the actual formatting of this information is defined in `apacite.sty`. If all four components are available, the default (APA manual) formatting of this looks like

*Acta Paedopsychiatrica*, 38(9), 229–231

but by redefining this command, users can change this, e.g., to

*Acta Paedopsychiatrica* **38**, No. 9: 229–231

The advantage of this is that users can fairly easily construct a customized style by redefining only a few commands in L<sup>A</sup>T<sub>E</sub>X, by using `\renewcommand`. Users will generally find it much easier to do this than adapt the `.bst` files. Furthermore, only the affected commands need to be redefined, so that, when a new version of `apacite` becomes available, all the new features can be used—with the exception of the redefined commands, of course—without having to adapt everything again. The drawback is that the `.bbl` file becomes less readable and more difficult to adapt: Previously, it was easy to make ad-hoc changes by editing the `.bbl` file, but this has now become more difficult. However, I intend to make this facility more user-friendly in the future, e.g., by introducing options in the L<sup>A</sup>T<sub>E</sub>X package (say, `boldvolume`) or by introducing user-friendly commands that accomplish the same (say, `\boldvolume`), so that the advantages will clearly outweigh the disadvantages for the vast majority of users.

Some packages (`amsrefs` and `biblatex` are the ones I am aware of) handle the formatting of the reference list completely in the L<sup>A</sup>T<sub>E</sub>X domain, using BibTeX only for sorting the reference list. The current version of `apacite` does not go that far; there are still certain choices that are hardwired in the BibTeX code. Examples are the ordering of the various blocks (most notably the position of the year; some styles have this at the end) and the formatting of names. It is doubtful whether `apacite` will follow the path toward the end, like these other packages. Some formatting choices are closely connected with sorting of entries (e.g., by title if there is no author; this assumes that the title then takes the author position) or other cross-entry relations (such as adding ‘a’ and ‘b’ suffixes to the year). Therefore, it does not seem useful to disconnect these choices by handing them over to L<sup>A</sup>T<sub>E</sub>X.

The following commands are currently implemented to format the reference list. If you want

to customize them, you can study the source code in the file `apacite.sty`, copy it to your own package file, and adapt the code as desired.

|                                            |                                                                |
|--------------------------------------------|----------------------------------------------------------------|
| <code>\APACmonth</code>                    | Month name (as function of a number)                           |
| <code>\APACrefYear</code>                  | Year in the reference list                                     |
| <code>\APACrefYearMonthDay</code>          | Year + month + day                                             |
| <code>\APACrefatitle</code>                | Title (article style, default upright)                         |
| <code>\APACrefbtitle</code>                | Title (book style, default italicized)                         |
| <code>\APACrefaetitle</code>               | English translation of title (article)                         |
| <code>\APACrefbetitle</code>               | English translation of title (book)                            |
| <code>\APACjournalVolNumPages</code>       | Journal, volume, number, and pages                             |
| <code>\APACaddressPublisher</code>         | Address and publisher                                          |
| <code>\APACaddressInstitution</code>       | Address and institution (for technical reports)                |
| <code>\APACaddressPublisherEqAuth</code>   | Address and publisher if the publisher is the author           |
| <code>\APACaddressInstitutionEqAuth</code> | Address and institution if the latter is the author            |
| <code>\APAChowpublished</code>             | <code>howpublished</code> field                                |
| <code>\APACorigED</code>                   | Original editor for reprints and translations                  |
| <code>\APACorigEDS</code>                  | Ditto, multiple editors                                        |
| <code>\APACrefnote</code>                  | Formatting of <code>note</code> field                          |
| <code>\APACorigyearnote</code>             | Original year + optional note                                  |
| <code>\APACorigjournalnote</code>          | Original journal (+ volume, number, pages) + optional note     |
| <code>\APACorigbooknote</code>             | Formatting of “original” information if the original is a book |
| <code>\APACbVolEdTR</code>                 | Technical report information                                   |
| <code>\APACbVolEdTRpgs</code>              | Ditto, for <code>intechreport</code>                           |

In addition, the following environments are used:

|                                |                                                                                                 |
|--------------------------------|-------------------------------------------------------------------------------------------------|
| <code>APACrefURL</code>        | Formatting of <code>url</code> field                                                            |
| <code>APACrefURLmsg</code>     | Formatting of <code>url</code> field for messages                                               |
| <code>APACrefannotation</code> | Formatting of <code>annote</code> or <code>annotate</code> fields for annotated bibliographies. |

It is likely that these lists will be expanded in future versions of `apacite`.

## 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, and because I write all my scientific work in English, `apacite` originally did not contain explicit support for other languages. However, most language-specific elements have been implemented in the form of  $\text{\LaTeX}$  commands, so that users could define their own  $\text{\LaTeX}$  package in which these elements were changed. The labels discussed in section ?? above are the most important part of this.

Furthermore, when writing in a non-English language, the `babel` package will usually be loaded. The `babel` package is an extensive package that facilitates using  $\text{\LaTeX}$  for documents in languages other than English. There have been some compatibility problems between `babel` and `apacite`, and therefore since 2003 `apacite` contained some explicit code to overcome the compatibility problems. I think this works well, provided that `apacite` is loaded *after* `babel`.

Thus, although `apacite` did not support non-English languages explicitly, it did facilitate them.

Supporting non-English languages is not a main objective of **apacite**, because it is not needed for the APA. Nevertheless, it potentially is a useful addition. 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 if, with a few adaptations, **apacite** can also be used in these circumstances. For this reason, and based on user requests, **apacite** has contained some explicit language support since June 2005.

The unpacked **apacite** distribution contains files that have names according to the construction  $\langle language \rangle.apc$ , i.e., **english.apc**, **dutch.apc**, etc. These files define the language-specific modifications of **apacite**, mainly translations of terms like “and”, “Ed.”, etc. They have been written by **apacite** users who are (native) speakers of the languages involved. 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. However, sometimes an **.apc** file makes some assumptions (such as **greek.apc**, which assumes ISO-8859-7 encoding) or makes some choices that are nontrivial. Therefore, look at the source code of the **.apc** file (or in the Implementation section of this manual if you have generated this manual with the `\AlsoImplementation` option) if you obtain unexpected results. These files contain some brief documentation of the specific issues relevant for the language at hand.

The list with available **.apc** files has been given in section ???. 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 supplied. If an **.apc** file is not available for the language you need, you can use one of the supplied ones as a template and write your own. If you send it to me, I can include it in a next version of **apacite**.

Note, however, that this form of language support is still very limited. There are many aspects that are not yet covered, like different forms of dates (month-day vs. day-month). As is clear from this manual, some facilities for expanded language support have been implemented in **apacite**, but this has often not yet been applied in the **.apc** files. Thus, future versions of **apacite** need to improve on this.

The **babelbib** package offers very sophisticated support of different languages. Perhaps some of the features that it offers are useful additions to **apacite** as well, and these may be included in a future version of **apacite**.

## 8 Compatibility

Generally speaking, I would want **apacite** to be compatible with other packages, of course. However, what does compatibility mean? It can mean various things:

1. The same **.bib** file can be used with different **BIBTEX** styles and **LATEX** citation packages;
2. The same citation commands as other citation packages;
3. Another **LATEX** citation package can be used with a **.bbl** file that is generated by the **apacite** **BIBTEX** style;
4. The **apacite** **LATEX** citation package can be used with a **.bbl** file that is generated by another **BIBTEX** style;

5. The `apacite` L<sup>A</sup>T<sub>E</sub>X citation package can be loaded at the same time as other packages without errors or undesirable side-effects.

Because of the complicated rules of the APA manual, it is not always possible to achieve all kinds of compatibility with all other packages. Later in this section, I will discuss compatibility issues with other packages with which known incompatibility problems exist or have existed. A first example of this, the `babel` package, has already been mentioned above in section ??.

Point ?? is the most important. If you have to write a completely different `.bib` file for different citation styles, there does not seem to be an advantage in using BibT<sub>E</sub>X. You might as well write the reference list explicitly in L<sup>A</sup>T<sub>E</sub>X 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;
- Journal names are abbreviated by some styles, but not according to the APA manual rules;
- Publisher names should be abbreviated according to the APA manual, e.g., “John Wiley & Sons” becomes “Wiley”. Other styles do not do this;
- The issue number of a journal should only be given if the journal starts each issue with page 1, not if page numbers of different issues in the same volume are consecutive. Other styles may require the issue number in all cases;
- Page ranges are given as “341–351” in APA style, whereas other styles use “341–51”;
- The APA rules require a capital after a colon (‘:’), so that subtitles start with a capital as well. Other styles may not do this;
- The APA has specific rules about the address: For a U.S. city, the state (or territory) should be given as a 2-letter code from the U.S. Postal Services, and the province and country should be given if it is a non-U.S. city, unless it is one of the 17 (U.S. and non-U.S.) cities that are “well-known for publishing”. Then, only the city should be mentioned. There are similar exceptions if the name of the state (and sometimes even city) is already mentioned in the university name and the university acts as publisher;
- Yearbooks like *Annual Review of Psychology* should be treated as journals according to the APA rules, whereas other styles treat these as books;
- The additional fields (such as `translator` and the `original*`-fields) that are used by `apacite` but are not defined in other BibT<sub>E</sub>X styles are of course `apacite`-specific.

Some of these problems can be handled relatively elegantly. For example, the user could use a command like “`\RomanVol{3}`” in the `volume` field. Then, the user could define a counter, `RomanVolcounter`, say, in L<sup>A</sup>T<sub>E</sub>X and then define `\RomanVol` (similar to `\BCnt` discussed earlier) as

```
\newcommand{\RomanVol}[1]{%
  \setcounter{RomanVolcounter}{#1}\theRomanVolcounter}
```

where `\theRomanVolcounter` is defined as `\arabic{RomanVolcounter}` if `apacite` is loaded and as `\Roman{RomanVolcounter}` if another package is loaded that desires this. The example with style-specific `.bib` files that contain the definitions of journal-strings was already given in section ?? above. However, many users will not go through all this trouble, and not all problems can be solved in this way. So we have to realize that not all entries in the `.bib` file will be suitable for all 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.

Point ?? above is currently not fulfilled. 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 `\citete` instead of `\cite` and `\citeA`, and uses many more alternative commands, see section ?? below. The `chicago` package uses `\citeA` instead of `\citeauthor`, many “numerical” citation styles only recognize `\cite`, and so forth. As discussed below, I now consider `natbib` as the standard with which other packages should comply. Therefore, future versions of `apacite` will support the `natbib` citation commands, but the current version does not do that.

I think that points ?? and ?? are not that important. It seems to me that there is not much wrong with defining a style through a combination of a `.sty` and `.bst` file, each requiring the other to work. However, given the possibilities and philosophy of the `natbib` package, it would be desirable that `natbib.sty` can be used with `apacite.bst`. As discussed below, this works, although it requires `apacite.sty` to be loaded as well, before `natbib.sty`.

Point ?? is 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 usefully be compared, and are more valuable when they can be used jointly. The rest of this section addresses this issue for a number of packages with which compatibility problems are known to exist or have existed.

## 8.1 natbib

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. I will mention some of these packages below.

I consider `natbib` as the *de facto* standard with which other packages should be compatible. Unfortunately, `apacite` is still not entirely compatible with `natbib`. The main incompatibility is that different citation commands are used, so that it is not possible to use the same document with `natbib` or `apacite`.

`apacite`, building on its predecessors, uses `\cite` for parenthetical citations and `\citeA` for in-text citations, whereas `natbib` uses `\citep` and `\citete` for these purposes. Moreover, text that should precede the citation is entered between < and > marks and text that should follow the citation is entered between square brackets ([ and ], i.e., a standard `LATEX` optional argument) in `apacite`, whereas `natbib` uses *two* optional arguments between brackets. If there is one optional argument, `natbib` interprets this as text following the citation (just like `apacite`). If there is text that

should precede the citation, but not text that should follow the citation, two optional arguments must be used in `natbib`, the second being empty.

Furthermore, both packages have defined some alternative citation commands, such as `\citeNP` (`apacite`) and `\citealp` (`natbib`), some of which serve the same purpose and could therefore be mapped onto each other easily, but some others have no equivalent in the other package.

A partial conversion table, listing the main `apacite` commands and their (closest) `natbib` equivalent is given in Table ?? . The APA requirement and `apacite` behavior of listing the full author list for the first citation (if there are at most 5 authors) and the short author list in subsequent citations is accomplished by loading `natbib` with the `longnamesfirst` option. This is assumed in the table. An example of better APA-like behavior of `natbib` is that, by using the `sort` option, `natbib` is able to sort the references within the same citation command, as required by the APA manual, whereas `apacite` does not have this option yet.

Table 3: Partial conversion table of the main `apacite` citation commands and their (closest) `natbib` equivalents.

| <code>apacite</code>            | <code>natbib</code>                        |
|---------------------------------|--------------------------------------------|
| <code>\cite</code>              | <code>\citep</code>                        |
| <code>\citeNP</code>            | <code>\citealp</code>                      |
| <code>\citeA</code>             | <code>\citet</code>                        |
| <code>\citeauthor</code>        | <code>\citeauthor</code>                   |
| <code>\citeauthorNP</code>      | <code>\citeauthor</code>                   |
| <br>                            |                                            |
| <code>\fullcite</code>          | <code>\citep*</code>                       |
| <code>\fullciteNP</code>        | <code>\citealp*</code>                     |
| <code>\fullciteA</code>         | <code>\citet*</code>                       |
| <code>\fullciteauthor</code>    | <code>\citeauthor*</code>                  |
| <code>\fullciteauthorNP</code>  | <code>\citeauthor*</code>                  |
| <br>                            |                                            |
| <code>\shortcite</code>         | <code>\shortcites{\keys}\citep</code>      |
| <code>\shortciteNP</code>       | <code>\shortcites{\keys}\citealp</code>    |
| <code>\shortciteA</code>        | <code>\shortcites{\keys}\citet</code>      |
| <code>\shortciteauthor</code>   | <code>\shortcites{\keys}\citeauthor</code> |
| <code>\shortciteauthorNP</code> | <code>\shortcites{\keys}\citeauthor</code> |
| <br>                            |                                            |
| <code>\citeyear</code>          | <code>\citeyearpar</code>                  |
| <code>\citeyearNP</code>        | <code>\citeyear</code>                     |
| <code>\nocite</code>            | <code>\nocite</code>                       |

Many labeling and punctuation commands are also different but similar between `natbib` and `apacite`. I haven't studied the `natbib` commands in detail, but will do so in the future. It is my intention to add `natbib`-compatible commands to `apacite` in the future, so that the same document can be processed either with `natbib` or with `apacite`.

When you want to use `natbib` for citation and `apacite` for the reference list, you still need to load the L<sup>A</sup>T<sub>E</sub>X package `apacite.sty` (with `\usepackage`), because of the commands that are included in the B<sup>I</sup>B<sub>T</sub>E<sub>X</sub> output (as defined in section ?? above). The `apacite.sty` file must be loaded *before* `natbib.sty`, otherwise you will get lots of error messages. I may try to fix that for a future version of `apacite`.

Using `natbib` for the citations has some advantages over using `apacite` for the citations. Apart from the abovementioned sorting of citations within a single citation command, these are mainly `natbib`'s better compatibility with some other important packages, although many compatibility problems with `apacite` have now been solved.

However, `natbib` does not fully comply with the APA rules. Known incompatibilities between `natbib` and the APA rules are: (1) Multiple adjacent citations with the same author and year are formatted as “Johnson, 1991a,b” by `natbib`, whereas the APA manual requires this to be “Johnson, 1991a, 1991b”; (2) The APA manual requires “and” between authors to be “and” in text and “&” in parenthetical citation, but `natbib` does not make this distinction.

I could change the `apacite.bst` and `apacitex.bst` BibTeX style files so that `natbib` would be “fooled” and the first problem would be resolved. However, this has the drawback of imposing APA style citations when you request `natbib` style citations, and you may actually want the behavior mentioned above. I am only able to provide one of these possibilities. It would be better to add an option to `natbib` with which the desired behavior can be chosen. Thus, you could try to put pressure on the author of `natbib` to achieve this. Alternatively, I might attempt to provide an option to `apacite` in the future that controls this behavior of `natbib`.

The second incompatibility between `natbib` and the APA rules mentioned above occurs because `apacite` uses the re-definable `\BBA` command for this usage of “and”. The `natbib` citation commands do not redefine `\BBA`, so you can choose only one of these, or you must redefine `\BBA` explicitly before each citation.

## 8.2 `hyperref`, `backref`, and `url`

The `hyperref` package turns (cross-) references into hypertext links. This can be used in conjunction with a program such as `LATEX2HTML` to compose `.html` files with clickable links, to internet pages or within the same document, or it can be used to create a `.pdf` document with clickable crossreferences. Evidently, citations are also references. Therefore, the `hyperref` package transforms these into hypertext links as well, and needs to redefine the citation commands and reference list commands in order to do so. These redefinitions have caused severe incompatibility problems between `apacite` and `hyperref` in the past. Previous versions of `apacite` contained an experimental `hyper` option that solved some problems, but left others. In the current version of `apacite`, the incompatibility problems have been solved, and the `hyper` and `nohyper` options are now obsolete. However, some of the “dirty tricks” I needed for complicated examples in section ?? still conflict with `hyperref`. You will probably not encounter these problems very often, but if you do these can be easily solved by editing the `.bbl` file instead of using dangerous tricks; see also section ?? below.

The `backref` package adds to each entry in the bibliography a list of pages (or sections) on which it is referenced. Hence, it serves as an alternative to an author index, in which not the entries themselves are backreferenced, but the authors of these entries, see section ?. Note that `natbib` offers yet another possibility, viz. the citations in the standard index, or a separate citation index, see the `natbib` documentation. In my opinion, the `backref` way of indexing the page numbers of the citations is more logical. Moreover, it is more condensed as well.

Obviously, the `backref` package has to change the citation commands and reference list to be able to do its backreferencing. Because `apacite` redefines these as well, they are conflicting to some degree. A similar compatibility problem between `backref` and `natbib` was noticed by the author of `backref`. Therefore, `backref` contains some code to make it compatible with `natbib`. This code has been adapted for `apacite` and incorporated in `apacite`. Thus, `backref` as a standalone package has been made compatible with `apacite`. The drawback of this solution is that if changes are made in



the `backref` package, this may result in new incompatibilities with `apacite` unless `apacite` is changed as well. Whether, and if so, when, this will occur will be a question for the future.

Apart from being a standalone package, `backref` can also be combined with `hyperref`. It is then not loaded separately, but through the `backref` or `pagebackref` option of the `hyperref` package. In this case, the backreferences become hypertext links. See the discussion of the `hyperref` package above.

The `url` package provides the `\url` command for formatting URLs, i.e., web addresses. It is typically loaded if `hyperref` is also loaded, but it is very useful on its own as well. There are no compatibility problems with `apacite` and it is not required when using `apacite`. However, if you use the `url` field, `apacite` puts it into a `\url` command. If the `\url` command is not defined by another package, `apacite` provides a crude definition, so that no errors are generated. But the formatting thus obtained usually does not satisfy the APA rules about formatting, especially its rules about line breaking within a URL. The default line breaking rules of the `url` package are fairly close to the APA rules, but there are some differences. Therefore, `apacite` provides several URL-styles that adhere to the APA line breaking rules, supplemented by the additional suggestions in ? (, p. 645). These styles apply to URL formatting through the `\url` command anywhere in the document, not only in the reference list and not only in the `url` field.

The only difference between the URL styles defined by `apacite` is which font family they use: `APACtt` uses `\ttfamily`, `APACsf` uses `\sfamily`, `APACrm` uses `\rmfamily`, and `APACsame` does not define a font family, and thus uses the font family that is active at the start of the URL. `APACtt` is selected by default, but you can select another URL style by inserting the command `\urlstyle{<style>}` in your document, after `\begin{document}`. See the documentation of the `url` package for more details.

### 8.3 Multiple bibliographies

The `chapterbib` package allows separate bibliographies for separate chapters of a book, or, technically, separate `\included` files. To work properly, it needs some small changes to the citation package, which are clearly described in the `chapterbib` documentation. These changes have been incorporated in `apacite`, and therefore `apacite` is compatible with `chapterbib`.

It is generally advisable not to use the `sectionbib` option of `chapterbib`, because this redefines `\thebibliography`, which interferes with the redefinition by `apacite`. You can use the `sectionbib` option of `apacite` to accomplish that the bibliography or bibliographies are sections, rather than chapters.

Moreover, `apacite` is configured such that it takes a section as default if it is in the main matter (signified by `\mainmatter`) of the document. In the back matter (`\backmatter`) the bibliography is put in a chapter. This is also the case if `\chapter` is defined but `\mainmatter` not. (Is this possible? Perhaps for reports?) Consequently, the `sectionbib` option is only useful in specific circumstances. The `nosectionbib` forces the bibliography to be a chapter, provided `\chapter` is defined. (I figured a `chapterbib` option would be a bit confusing, so I stuck with `nosectionbib`.)

The `bibunits` package is an alternative to `chapterbib`. It allows a greater flexibility in choosing the scope of a bibliography (chapter, section, or arbitrarily defined part of the document). In previous versions, there have been some compatibility problems between `apacite` and `bibunits`, but these have now been solved.

The `multibib` and `multibl` packages support using multiple bibliographies simultaneously. Thus, one can cite entries from multiple bibliographies in the same part of the document. This may be useful if one wants thematically different bibliographies or separate bibliographies for references in

different scripts (e.g., Latin and Greek). These packages accomplish this by introducing different citation commands for the different bibliographies. **apacite** contains compatibility code such that these packages can be used jointly with **apacite**.

The **bibtopic** package offers a similar functionality, but instead of using separate citation commands, it requires the user to keep the references in different **.bib** files, so that there is one bibliography per **.bib** file. **apacite** contains a little bit of compatibility code, which largely seems to work, but “and” between authors is always “&”, even in `\citeA`. I hope to fix this in a future version of **apacite**.

Another package that offers similar functionality is **splitbib**. This uses neither different citation commands, nor separate **.bib** files. rather, it requires the user to declare the “category” of each cited reference, typically at the beginning of the document. Again, **apacite** contains compatibility code such that simultaneous redefinitions of some of the citation and reference list commands do not conflict.

A few caveats must be mentioned here. First, simultaneous bibliographies only work well if it is guaranteed that there are no ambiguities in the reference lists. With a single bibliography, ambiguities are resolved at the **BIBTEX** stage by adding initials to ambiguous first author names, adding additional authors before (or instead of) “et al.” to ambiguous multiple-author lists, and adding “a” and “b” suffixed to the year for references to different documents with the same authors published in the same year. Because, with the exception of **splitbib**, these packages require multiple **BIBTEX** runs on separate selections of references, **apacite.bst** can only resolve ambiguities within the same bibliography. When there are cross-bibliography ambiguities, these are not resolved. It seems like these packages are primarily intended to work with numbered citation styles, or otherwise labeled citation styles, so that unique cross-bibliography numbers or labels can be generated at the **L<sup>A</sup>T<sub>E</sub>X** stage.

Second, I wrote the compatibility code in early 2006 (sorry about the long delay for this release) and have only superficially tested it recently. It seems to work, apart from minor details, but there may be problems I am not aware of. Please let me know if you encounter such problems. Then I’ll try to fix them or at the least mention them in the next version of the manual. In particular, I believe that usage of `\nocitemeta` is not perfectly integrated yet: You may get the “References marked...” message even if the relevant bibliography does not contain marked references, or references that are `\nocitemeta`’d for one bibliography but not for another are marked for both. I don’t think the `\nocitemeta` command is used much, so I would be surprised if anybody encountered such problems, but please let me know if you do. Then I’ll try to fix them.

Third, the compatibility code for all of these packages either follows instructions given in those packages, or brutally redefines part of the code of those packages. If these packages change, this may mean that new compatibility problems are created, or that improved features of these packages are not available when combined with **apacite**. I do not have the time to regularly check for updates of these packages and then study changes and/or test compatibility again. Instead, I only take action when a user notifies me of a problem, or I coincidentally find out myself. Therefore, again, please let me know if you encounter problems.

## 8.4 Programs for conversion to html, rtf, etc.

There are several programs that convert **L<sup>A</sup>T<sub>E</sub>X** files to other types of files, most notably **html** and **rtf**. This may, for example, be useful if you want to publish your document on the internet or if you are requested to send a “Word” file to a journal. I have received reports that **L<sup>A</sup>T<sub>E</sub>X2HTML** and **L<sup>A</sup>T<sub>E</sub>X2RTF** do not handle documents that use **apacite** well. Therefore, I have studied these

programs. This section contains my observations. Note that these are quite old and not updated, and it is quite likely that the software mentioned in this section has been altered (improved) considerably. Therefore, the information may be outdated. Please let me know if there are major changes that I should be aware of.

`LATEX2HTML` is a *Perl* program that converts a `LATEX` input file into one or more `.html` files, so that a document that is prepared with `LATEX` can be used as a (user-friendly) internet document. For papers, lecture notes, and transparencies of classes, I always convert the `.dvi` file output by `LATEX` to a `.pdf` file (through the `dvipdfm` program), which can be read by most people, e.g., using the Adobe Reader program or browser plug-in. I find `.pdf` files more convenient, because I can download a whole paper or presentation with a single click and then print it and read it when and where I desire, whereas typical examples of conversions to `.html` use many `.html` files, which can only be conveniently read from the computer screen when and where you're connected to the internet. Nevertheless, I can imagine that someone would like to convert a document to `.html`, e.g., because it loads faster and reads easier from the screen, because for some documents most people will have to use only a small part and can easier browse or search through it, or just because you want to make a home page and do not want to learn the html-language.

Similarly, `LATEX2RTF` is a standalone program that converts a `LATEX` document to `.rtf` format, which can be read by most word processors. This may be convenient if you (a `LATEX` user) want to share a document with someone who does not work with `LATEX`, or as mentioned above, if you are required to send a “Word” file to a journal.

Both `LATEX2HTML` and `LATEX2RTF` implement `LATEX` macros etc. in another programming language (`LATEX2HTML`: *Perl*; `LATEX2RTF`: C) and process the `.tex` files directly. This design implies that the conversion program must know the definitions of all `LATEX` macros that are used. A drawback of this design is that essentially the same code has to be written in different programming languages (`LATEX` macros for the `.sty` file, *Perl* for `LATEX2HTML`, and C for `LATEX2RTF`), but typically with some adaptations specific for the application at hand.

With `LATEX2RTF`, the `LATEX` macros that it is able to process are included in the C source code that is compiled. Consequently, no user additions are possible in principle. However, the source code is freely available, so if you define a `LATEX` macro, you could adapt `LATEX2RTF`'s source code and recompile `LATEX2RTF`. It is obvious that this is very inconvenient, especially if `LATEX` macros are defined differently for different styles or if they are defined differently in different parts of the same document.

Version 1.9.15 and higher of `LATEX2RTF` contains some support for `apacite`, but lags a few versions of `apacite` behind and thus doesn't work with recent versions of `apacite`. Moreover, even if this would be fixed, e.g., because I (or you) send patches to the `LATEX2RTF` authors, this wouldn't do justice to the flexibility of `apacite`, i.e., its customization options.

In contrast with this, `LATEX2HTML` allows the possibility to supply external *Perl* scripts. If `LATEX2HTML` encounters a `\usepackage{⟨package⟩}` command in the document, it attempts to read the corresponding `⟨package⟩.perl` *Perl* script. `LATEX2HTML` is shipped with many such *Perl* scripts. There is not one for `apacite`, but there is one for `natbib`, which could possibly be adapted for `apacite`. Alternatively, you could use the `natbib` citation commands and use the `natbib` package, which should work. This will become more straightforward when, in a future version, `apacite` also supports the `natbib` citation commands. Then the `natbib.perl` script may possibly be copied to `apacite.perl` without having to adapt much. However, I have never studied *Perl*, and I do not particularly like the idea of having to learn that language for this purpose, so it is not very likely that in the near future `LATEX2HTML` will work well with `apacite`.

Note, however, that the design with the *Perl* scripts has the advantage that, if you customize

the usage of `apacite` by writing your own `.sty` file that renews the definitions of some macros, you can also supply a corresponding *Perl* script, so that your customizations are processed.

An alternative to both `LATEX2HTML` and `LATEX2RTF` is `TEX4ht`. This is a collection of programs and style files that convert `LATEX` to various formats, such as `html`, `xml`, and `OpenOffice` format. If you have `OpenOffice`, the latter can be used to convert to `Word` (or `rtf`) format. The principle of `TEX4ht` is to let `LATEX` do most of the formatting and processing of commands, and do the conversion at low-level `TEX` commands. In this way, all newly defined commands and user-defined changes are automatically supported. This even carries over to redefinitions of macros in the middle of a document. However, for some specific purposes, there are some additional things that must be arranged. Therefore, `TEX4ht` contains a large list of `.4ht` files that are necessary to support various `LATEX` packages. One of these is the `apacite.4ht` file. The main purpose of this file is to make citations into hyperlinks to the corresponding entries in the reference list. Currently, this only works partially, but the citation text itself is formatted correctly, so this is only a relatively minor inconvenience. But, of course, it would be better if this would work perfectly. Perhaps I'll study the problems in the future and solve this for a future version (although, technically, it is a `TEX4ht` problem and not an `apacite` problem).

The choice of which converter to use will typically not (only) depend on how good it works in conjunction with `apacite`. Furthermore, apart from the converters mentioned above, there are several additional converters that I haven't studied at all. I will only study them if I receive reports about incompatibility and even then it is far from certain that I will be willing or able to fix the problems. It would be very demanding if I would have to define the `apacite` commands in a large number of programming languages for a large number of different converters. Because I prefer the design of `TEX4ht` and because this converter seems to work best at the moment, I will give a bit more attention to this one.

## 9 Generating an author index

The `apacite` package contains an option to (almost) automatically generate an author index. This is done by using the `apacitex.bst` `BIBTEX` style file instead of `apacite.bst` and loading the `apacite.sty` `LATEX` package with the `index` option, either explicitly or implicitly through one of the `stdindex`, `tocindex`, `emindex`, or `ltxemindex` options.

In previous versions of `apacite`, the `index` option automatically loaded the `index` package for multiple indexes, so that you can have a subject index as well as an author index. However, the current document uses the standard `ltxdoc` class for documenting `LATEX` packages, and this class uses another method for supporting multiple indexes, which appears incompatible with the `index` package. I have yet been unable to resolve these compatibility problems and, therefore, have implemented another way to generate an author index in addition to other indexes. This uses the same principles as `ltxdoc` and thus does work for this user's manual. As a consequence, `apacite` now offers options that generate the author index either with the `index` package or without it.

The `indexpackage` option implies that author indexing is done according to the `index`-scheme, whereas the `noindexpackage` uses the `ltxdoc`-based scheme. These options are ignored if the `index` option of `apacite` is not requested. When the `index` option is requested, the `indexpackage` option loads the `index` package. This is the default unless it is disabled by using either the `noindexpackage` option or the `ltxemindex` option.

Obviously, the `index` package must be available in a directory that is read by `LATEX` if the `indexpackage` option is requested, either explicitly or implicitly. If a subject index is desired, it must be defined by the user, it is not defined by `apacite`. See the manual of the `index` package on

how to do this. A default author index *is* defined by `apacite`. The author index is then requested by putting

```
\printindex[autx]
```

at the point in the  $\text{\LaTeX}$  document where the index is supposed to appear. To include the author index in the processed document, the following sequence must be followed: First,  $\text{\LaTeX}$ , which puts citation entries in the `.aux` file. Second,  $\text{\BibTeX}$ , which generates the bibliography in the `.bbl` file. Third,  $\text{\LaTeX}$  (at least) twice, which puts index entries in a `.adx` file, Fourth, *MakeIndex*, which uses the `.adx` file as input and puts the formatted index in a `.and` file, which can be processed by  $\text{\LaTeX}$ . So, if the main document is `filename.tex`, then *MakeIndex* must be called as follows:

```
makeindex -o filename.and filename.adx
```

although the command with which the *MakeIndex* program must be called may on some systems be slightly different, e.g., `makeinde` or `makeindx`. Furthermore, several additional options may be chosen on the *MakeIndex* command line. Fifth, run  $\text{\LaTeX}$  again, which (finally) includes the formatted author index in the `.dvi` file.

The `apacite` package provides some options to change the appearance of the index somewhat. With (only) the `index` option, the indexing facility is turned on, but the `theindex` environment is not explicitly (re)defined by `apacite`. This means that this environment is defined just like in standard  $\text{\LaTeX}$ , or the `index` package, or some other package that has been loaded that redefines this environment.

The `stdindex` option explicitly uses the definition of the `theindex` environment that is defined in the `index` package [1995/09/28 v4.1beta Improved index support (dmj)]. 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. 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. The `ltxemindex` option uses the same style as the `emindex` option, except that it does not load the `index` package and is thus tailored to a somewhat different indexing method.

When a table of contents entry is desired, another additional  $\text{\LaTeX}$  run is required to obtain the table of contents with the author index included. In exceptional cases, the extra line in the table of contents may cause the text to be moved as well, so that it may be necessary to run *MakeIndex* a second time, followed by yet one or two more  $\text{\LaTeX}$  runs.

The definitions of the index discussed apply to all indexes in the document, so to, e.g., the subject index as well as the author index. This may be undesirable, and therefore `apacite` provides the following commands to switch index styles:

```
\APACstdindex
```

```
\APACtocindex
```

```
\APACemindex
```

```
\APACltxemindex
```

These correspond to the package options `stdindex`, `tocindex`, `emindex`, and `ltxemindex`, respectively. Note, however, that the `ltxemindex` option and/or the `\APACltxemindex` command

generally cannot be used in a document that uses one of the other styles, and vice versa, because the way `index` works is incompatible with how standard  $\text{\LaTeX}$  indexing works.

If you want a different index style, either for all indexes in your document or only for some indexes, you can define such a style using the `\renewenvironment` command in the appropriate place. See the standard  $\text{\LaTeX}$  documentation for this command. The source code of the four commands mentioned here can be used as a template for such a style definition.

Finally, some elements of the formatting of the index itself can be changed by using (and/or writing) an `.ist` file containing the preferred options. *MakeIndex* should then be called with the filename of this index style file after a “-s” option on the command line, so, e.g.,

```
makeindex -s mystyle.ist -o filename.and filename.adx
```

The *MakeIndex* documentation discusses the possible options that can be put into the `.ist` file.

As discussed in sections ??, ??, and ??, the `suppresscorporate` and `includecorporate` options can be used to either excludes corporate authors from the author index or include them in it. The `\bibcorporate` command must be used in the `.bib` file to denote a corporate author.

For some of the examples from the APA manual, I had to use quite advanced  $\text{\LaTeX}$  tricks to obtain the correct formatting in the text, the reference list, and the author index; see section ?? below. This will typically not be necessary in practice. In the unlikely event that you encounter similar difficult cases, it will usually be sufficient to use a relatively simple trick in the `.bib` file. In rare situations, this may still give incorrect formatting, most likely in the author index. Somewhat less rare is the situation when there is such a problematic case but you also want to use the `hyperref` package. The tricks I used conflict with `hyperref`.

An easy solution to incorrect formatting and/or sorting in the author index is to edit the `.bbl` file manually. This should be done at a time when  $\text{\BibTeX}$  will not have to be run again. Otherwise, the changes would be overwritten by the next  $\text{\BibTeX}$  run. This editing of the `.bbl` file is not in the  $\text{\TeX}$ -spirit, but in my experience—I used it for two books with lots of references (?, ?, ?), with a previous version of `apacite` that caused a fair amount of 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 connection between the `.bbl` file and the author index is through `\AX` commands that are entered by `apacite.bst` in the `.bbl` file. These are responsible for the author indexing facilities. By looking at the contents of the `.bbl` file, some changes that lead to correct formatting are immediately obvious. For example, the entry for Bender is generated by the following lines:

```
\AX{bender jdirector
@Bender, J\BPBI
{\protect\biblital{Director}}\protect\indexskipbracenodot}.}%
```

It is immediately clear that this can be changed to

```
\AX{bender j
@Bender, J.}%
```

Then, this author will be formatted correctly in the author index, and conflicts with `hyperref` are resolved. Note that `apacite` uses the  $\langle key \rangle @ \langle visual \rangle$  form of indexing. This form is used in *MakeIndex* to distinguish between the actual representation in the index and the place where it should occur. The  $\langle key \rangle$  is used by *MakeIndex* to sort the entry, but the entry actually appearing in the index is  $\langle visual \rangle$ . In the  $\langle key \rangle$  part as formatted by `apacite`, all names are in lower case letters and all accents and punctuation are removed. Furthermore, `apacite` inserts *two* spaces between

surname and initials, and to *MakeIndex*, two spaces are different from one space. Spaces between surnames are removed, as well as spaces between initials. Finally, note that **apacite** always inserts a new line before the “@”. This is to avoid bad line breaks by **BIB<sub>T</sub>E<sub>X</sub>**. To *MakeIndex*, this means that each citation  $\langle key \rangle$  ends with a space.

The author index of the current document does not contain entries for the authors Shocked and Goodenough as cited in section ??, although it does for the reference list part. The reason for this is that the **key** field was used for the citations. If you use the **key** (and **firstkey**) field and you want the authors in the author index as well, you have to include the index command **\AX** explicitly. For example, the **key** field for Michelle Shocked is

```
key = {{\protect\bibsong{Shocked}{1992}{Over the
      Waterfall}{track~5}}},
```

and if we change this into

```
key = {{\protect\bibsong{Shocked\AX{shocked m
      @Shocked, M.}}{1992}{Over the Waterfall}{track~5}}},
```

then citations to her are included in the author index.

For corporate authors, the **\corporateAX** command is used instead of the **\AX** command. Depending on whether the **suppresscorporate** option or the **includecorporate** option is requested, either explicitly or implicitly, **\corporateAX** either swallows its argument or is equivalent to **\X**. To make sure that this ends up correctly in the author index, it is important that this command as entered in the **key** and/or **firstkey** field is exactly as output by **apacite.bst** for the **author** field. The easiest way to accomplish this is to start with **key** and/or **firstkey** fields that do not have the **\corporateAX** command yet. Then, run **L<sup>A</sup>T<sub>E</sub>X** and **BIB<sub>T</sub>E<sub>X</sub>** and extract the correct **\corporateAX** command from the .bbl file and insert this in the **key** and/or **firstkey** fields. Rerun **BIB<sub>T</sub>E<sub>X</sub>** and after that proceed as usual. For example, the following is an excerpt from an entry for one of the examples from section 3.96 in the APA manual:

```
author   = {{\bibcorporate{National Institute of Mental Health}}},
firstkey = {{National Institute of Mental Health [NIMH]}},
key      = {{NIMH}},
```

If I run **L<sup>A</sup>T<sub>E</sub>X** and **BIB<sub>T</sub>E<sub>X</sub>** on the current document, the .bbl file contains the lines

```
\corporateAX{nationalinstituteofmentalhealth
@{\bibcorporate{National Institute of Mental Health}}}%
```

Note the **\bibcorporate** command and the additional braces. If these are left out,

```
firstkey = {{National Institute of Mental Health
      [NIMH]}\corporateAX{nationalinstituteofmentalhealth
      @{National Institute of Mental Health}}},
key      = {{NIMH}\corporateAX{nationalinstituteofmentalhealth
      @{National Institute of Mental Health}}},
```

then this organization gets two separate entries in the author index, which is incorrect. Instead, if we copy them literally,

```
firstkey = {{National Institute of Mental Health
      [NIMH]}\corporateAX{nationalinstituteofmentalhealth
      @{\bibcorporate{National Institute of Mental Health}}}},
key      = {{NIMH}\corporateAX{nationalinstituteofmentalhealth
      @{\bibcorporate{National Institute of Mental Health}}}},
```

then everything is as desired. The additional spaces before the “@” sign make no difference, but percentage signs should not be included in the `.bib` file at the end of lines to hide a line break (as in  $\text{\LaTeX}$ ), because  $\text{\BibTeX}$  does not recognize the percentage sign as a comment. Rather, it includes it in the `.bbl` file and sometimes puts additional code behind it, which would then not be read by  $\text{\LaTeX}$  because  $\text{\LaTeX}$  does interpret the percent sign as a comment. This behavior has now become less likely, because `apacite` inserts numerous line breaks in the `.bbl` file to avoid this problem and, more importantly, line breaks inserted by  $\text{\BibTeX}$  at undesired locations (e.g., in the middle of a command name or in the argument of a `\r1` command), but is still a possibility.

## 10 Annotated bibliographies

Annotated bibliographies can be obtained by requesting

```
\bibliographystyle{apacann}
```

instead of

```
\bibliographystyle{apacite}
```

or

```
\bibliographystyle{apacannx}
```

instead of

```
\bibliographystyle{apacitex}
```

The  $\text{\BibTeX}$  style files `apacann.bst` and `apacannx.bst` are almost identical to `apacite.bst` and `apacitex.bst`, respectively. The only relevant difference is that these styles include the contents of the `annotate` or `annotation` field in the output, thereby generating an annotated bibliography.

According to ? (? , pp. 810–811), the `annotation` style uses the `annotate` field for annotations, whereas the `annotate` style defines the `annotation` field for this purpose. To be compatible with both, `apacann` and `apacannx` support both of these fields. However, they are not both included in the output for the same reference, in case someone wants to make a file that is compatible with both the `annotate` and `annotation` styles and thus duplicates the information. In such a case, the contents of the `annotate` field is included in the output.

The annotation is inserted in the `.bbl` file within a `APACrefannotation` environment. Hence, its formatting can be changed by the user by redefining this environment.

## 11 apacdoc.sty

Section ?? below cites all examples from the APA manual. This is a test of `apacite` and a proof that it does what it’s supposed to do—format citations and reference lists complying with the APA rules. However, the APA manual discusses many nonstandard issues and several of them need nonstandard treatment. Some of these can only be formatted by what I call “tricking” the system. This has been done by defining several  $\text{\LaTeX}$  commands that are too specifically tailored for a certain reference, or ad hoc in other ways, or more generally experimental and not well tested, and likely to lead to more (compatibility) errors. Therefore, I do not consider these commands suitable for inclusion in the `apacite` package proper, i.e., the file `apacite.sty`.



They are not important enough for that, and thus you would rarely need them, and for most of them I anticipated (or hoped) at one point that better solutions would be available in the future. Thus, it seemed quite likely that major changes in these commands would be made, and I didn't want to feel obliged to keep supporting them for backwards compatibility. Rather, I considered these as illustrations of the power of L<sup>A</sup>T<sub>E</sub>X and templates that users can base their own tricks upon. Accordingly, these commands were defined in the preamble of previous versions of the user's manual, which used to be the file `apacite.dtx`.

I still largely feel the same way about these commands, although I'm not convinced anymore that within reasonable time I will come up with elegant solutions that make them obsolete. Nevertheless, I still make improvements in this direction, and the support of the `url` and `lastchecked` fields in his version of `apacite` is an example of this.

With the restructuring of the distribution in a `.dtx` file, I have reconsidered the position of these commands, and have decided to put them in a supplementary package file, `apacdoc.sty`. As this name indicates, it is a package that is intended to be used for the `apacite` documentation (i.e., the current document).

It contains several commands and settings (like page margins) that are not interesting outside this document. However, as mentioned above, I needed many of the commands defined in this file for the examples in section ???. Thus, you might encounter similar situations as well, and these commands may be useful for you.

Because most of the commands in `apacdoc.sty` will not be useful for you (or may even interfere with your document), and because the settings defined in it will most likely not be what you want, it is generally not a good idea to load this file with `\usepackage` to use the commands that are useful to you. Instead, if you want to use them, you can copy their definitions to the top of your own document or personal style file (say, `mysettings.sty`, study their behavior, and perhaps improve them. This reflects the "illustration" or "template" nature of these commands.

Accordingly, this section briefly mentions the commands that may be useful outside the context of the current document. Thorough documentation is outside the scope of this manual; their definitions can be studied in the source code of `apacdoc.sty`, and their usage is illustrated in section ???. I will mildly support them, though: If you want to use them, you can ask me questions, but I will not put as much effort in fixing possible problems as I do with proper `apacite` issues.

The list below mentions (some of) the numbered examples in section ??? in which the commands are used.

`\bibleciteA`

`\fullbibleciteA`

`\shortbibleciteA`

`\biblecite`

`\fullbiblecite`

`\shortbiblecite`

`\bibleciteNP`

`\fullbibleciteNP`

`\shortbibleciteNP`

Citing the Bible. One mandatory argument: relevant chapter and verse, e.g., `\biblecite{1~Cor.\ 13:1}` (ex. 3.100-3).

`\bibleversion`

Defines the version of the Bible used: `\bibleversion{Revised Standard Version}`. Use this before using the bible citing commands, typically in the preamble of your document. Note that the Bible is not included in the reference list. You can use these commands as templates for references to other “classic” works. See ex. 3.100-3.

`\citesong`

`\fullcitesong`

`\shortcitesong`

`\citeAsong`

`\fullciteAsong`

`\shortciteAsong`

`\citesongNP`

`\fullcitesongNP`

`\shortcitesongNP`

These cite a song as required by the APA (except for the relative placement of closing quotes and comma). See ex. 69.

`\bibsong`

For `\citesong` and its siblings, the **key** field (and possibly **firstkey**) should contain the relevant information in the form of a protected `\bibsong` command:

```
key = {{\protect\bibsong{author}{year}{song title}{track info}}},
```

See ex. 69.

`\bibInstring`

This command has one argument. The macro changes the definition of the macro `\BIn` to the argument and then immediately changes the definition back to the old definition. So it makes a one-time change. Used to change “In” a book to “On” a CD for citing songs. See ex. 69.

`\BOn`

“On”, for citing songs “On” a CD. See ex. 69.

`\DSMcite`

`\DSMfullcite`

`\DSMshortcite`

`\DSMciteA`

`\DSMfullciteA`

`\DSMshortciteA`

`\DSMciteNP`

`\DSMfullciteNP`

`\DSMshortciteNP`

Commands to cite the DSM. See ex. 29. Can also be used for other “named” works. `natbib` offers a similar functionality through the definition of an *alias*, but I think that to get this working in the way the APA wants (i.e., the alias is not used for the first citation, but only for second and subsequent citations), it cannot be directly used. But the technical implementation of its functionality may form a useful basis for a better (more elegant) solution for this type of citation.

`\bibDSM`

For `\DSMcite` and its siblings, the `key` field (and possibly `firstkey`) should contain the relevant `\bibDSM` command:

```
key = {{\bibDSM{American Psychiatric Association}{Diagnostic  
and Statistical Manual of Mental Disorders}{4th  
ed.}\hbox{}}{DSM-IV}}},
```

See ex. 29.

`\DSMauthor`

`\DSMfulltitle`

`\DSMedition`

`\DSMshorttitle`

These four commands extract the four elements of a `\bibDSM` command. They are used in the `\DSMcite` command and its siblings, but may occasionally be useful by themselves.

`\bibliteral`

This command is used in some of the examples in section ?? to put some literal text into the reference list that would otherwise be abbreviated to a single “initial” at the `BIBTEX` stage. For example, in ex. 4.08, the authors are “Bulatao, E. (with Winford, C. A.)”, where “(with Winford, C. A.)” is inserted by wrapping it in a `\bibliteral` command, which itself is wrapped in an additional pair of braces, so that `BIBTEX` treats it as a complicated “accent” and the argument as a complicated single character. This text should not be put in the author index, and therefore some manipulation through redefining the meaning of this command before and after the reference list is used to suppress this text, as well as an additional period, there. Used in examples 4.08, 65, 66, 68, and 70.

`\bibinitial`

This command is similar to `\bibliteral`. It is used in one example (ex. 67), where the “author” is a “(Producer)”, but the initial of the author is put in the text because there is another author with the same surname (Miller). This causes problems not encountered in the examples where `\bibliteral` is used, and after a lot of trial and error, the current definition of `\bibinitial` was developed to format this particular author correctly in text, reference list, and author index. It looks like this is an improved version of the `\bibliteral` command, so perhaps in the next release I’ll use `\bibinitial` for all the examples in which now `\bibliteral` is used.

### `\bibreftext`

This command takes two arguments. By default, it ignores the first and inserts the second in the output. In this manual, it is redefined before the reference list to ignore its second argument and insert the first argument in the output. This is used in ex. 17, where the year in the in-text citation should be “1992/1993”, whereas in the reference list, it should be 1992 (and the 1993 is listed elsewhere). For reprints and translations, this is handled automatically by `apacite` if you use the `original*` fields, but this one is different.

### `\Dutchvon`

This command is equivalent to `\bibreftext`, but is used for a different situation, and therefore, depending on language, subtle style differences, and other preferences, you might want to change the definition of one but not of the other. This one illustrates how the Dutch way to treat “von” parts of surnames can be handled. It is discussed in ex. B-2, although it is not actually used in the examples, because the Dutch way to treat “von” parts is not APA style.

### `\SwapArgs`

This command takes two arguments and swaps their order. This is useful if you cite a work whose title, (corporate) author, or other element starts with an article that should be disregarded for ordering. `apacite` disregards “a”, “an”, and “the” in the sorting of titles, but these are obviously English. For French, you could say

```
title = {\SwapArgs{Marseillaise}{La }},
```

which is thus alphabetized under M. (Not used in the examples.)

### `\swallownext`

Eats up the next character. Dangerous but sometimes useful. Used in the definitions of some of the following commands.

### `\swallowdot`

Eats up the next character, but only if it’s a dot (‘.’). Used in the definitions of some of the following commands.

### `\bibnodot`

Used as `\bibnodot{.}`. Eats the period, but prevents `BiBTeX` from adding one. Used in examples 78, 79, 87, 91, and 95. This is likely to be useful for many users, and is stable, so it may be upgraded to `apacite.sty` in the next release of `apacite`. The only thing against this is that it is used in highly unelegant manual formatting of a fairly large piece of the description of a reference to a website in a `howpublished` field. I’d prefer to avoid this, but that would require either adding another idiosyncratic field or even less elegant trickery in the `.bib` file.

### `\bibskipbracenodot`

Suppresses the period after a subsequent closing brace. Quickly leads to errors because it uses very low level `TeX` code that (locally) changes some fundamental settings. This is used in some examples (65, 68) where `\bibliteral` is used to insert a description of a first “author”, such as “(Producer)”. This is interpreted by `BiBTeX` as a complicated one-letter first name, and thus it adds a period to make it into an initial. This period is the one that’s eaten by this command. Used in examples 65 and 68.

### `\indexskipbracenodot`

This serves the same purpose as `\bibskipbracenodot`, and it uses the same code, but whereas `\bibskipbracenodot` is activated for the reference list, `\indexskipbracenodot` is only activated for the author index. This has technical reasons and it does not work under all circumstances. Used in examples 4.08-3, 65, 66, and 68.

### `\bibeditortype`

This has one argument. The macro changes the definitions of the macros `\BED` and `\BEDS` to the argument and then immediately changes the definitions back to their old definitions. So it makes a one-time change. This is used when there is someone in an “editor-type” role, but with another designation, such as “producer” or “director”. Ex. 36, 67, and 68.

### `\renewbibeditortype`

The above-mentioned definition of `\bibeditortype` is only relevant in the reference list: In the citations in the text and in the author index, the command should do nothing. Therefore, the command is initially defined as “nothing”. The definition discussed above is initialized by `\renewbibeditortype`. This command is executed just before the reference list in the current document. (After the reference list, `\bibeditortype` becomes “nothing” again.)

### `\BAbstractRetrieved`

This command is similar to `\BRetrieved`, except that it inserts “Abstract retrieved” instead of “Retrieved”. Used in examples 82 and 83.

### `\BmakeAbstractRetrieved`

This command functions much like `\bibeditortype`: It replaces the next instance of `\BRetrieved` by `\BAbstractRetrieved`. Used in examples 82 and 83.

### `\noedition`

Suppresses the “ed.” (or “`\BED`”, rather) of an `edition`. Used in examples 92 and 94 to format a string like “Version 4” as an `edition`.

### `\APACmanBeforeBib`

Illustrates how some of the commands defined above are used in practice: This command is given just before the `\bibliography` command, and redefines some of the commands above, so that their meaning is different in the reference list from their meaning in the main text. Similar commands: `\APACmanBeforeIndex`, `\APACmanAfterIndex`.

## 12 Known problems and todo-list

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. These subjects will be discussed in this section.

A class of problems was already discussed in section ?? above. This concerns the incompatibility of the `.bib` file with other citation styles, through `apacite`-specific fields, APA-specific contents of fields, or through special tricks. These lead to correctly formatted references in APA style, but may cause problems with using the `.bib` file with other styles (in case APA journals reject your article ...). As discussed there, however, this problem is unavoidable and presumably not a very big problem to most users.

An incomplete list of other known problems and things that I would like to do with `apacite` follows below. Several items of this list are the direct result of the tricks I had to use for the examples in section ?? below.

- Support `natbib` citation commands (`\citet`, `\citep`, etc.).
- Redefine citation commands (and other commands) as “robust” commands so that they can be used in other commands, such as `\caption`, without having to use an explicit `\protect`. The `natbib` package does this, but `apacite` still adheres to the older “fragile” setup of its predecessors.
- Sort entries within a single citation command. The `natbib` package already offers this option.
- Swap (optionally) the order of closing quotes and various other punctuation marks (`.`, `,`, `!`, `?`) in a citation, for articles where there is no named author and the title (between quotes) takes the role of the author.
- Add an option that disables `apacite`’s citation commands (and perhaps a few others) to further improve inter-operability with `natbib`.
- Add options and/or commands for easier customization of reference list and author index.
- Continue with transferring reference list formatting code from `BIBTEX` to `LATEX`.
- Introduce options for the `BIBTEX` part, e.g., `index` vs. `noindex` (so that `apacitex.bst` becomes superfluous); full first names instead of only initials, or perhaps options for putting initials in front of (2nd and later) author names; unsorted reference lists or lists sorted chronologically (forward or reverse; for CVs); etc.
- Give explicit support for citation to the Bible and other “classic” works and the *DSM*.
- Give explicit support for music recordings in `incollection` entries.
- Perhaps study referencing legal materials (Appendix D of the APA manual). This is actually not high on my priority list. Perhaps you can use the `camel` package if you do this often.
- Fix a bug (which is unlikely to show up) in the `.bst` file: If there are two or more references, published in the same year, each with 6 authors, the first 5 of which are the same but the 6th authors are different, then these lead to ambiguous citations. The same thing happens when one has 6 authors and another 7, with the same first 6 authors.
- Extend the `\bibcorporate` command with an optional argument that gives the abbreviated form, so that the `firstkey` and `key` fields do not have to be used for this common situation.
- Support nonstandard types of authors, like “Director” and “Series Ed.”. This currently must be tricked in a nonelegant way. Perhaps I (or you) can think of a better solution.
- If a work is authored by someone, “with” someone else: This currently must be tricked in a nonelegant way. Perhaps I (or you) can think of a better solution.
- Define `BIBTEX` macros or strings for commonly encountered journals (at least the APA journals).
- In an `unpublished` entry, if `author` and `organization` are equivalent, replace the latter with “Author” in the output, analogous to the situation where `publisher` or `institution` are equal to `author` in several of the other entry types.

- Allow `lastchecked` fields of the form 2009-02-01 and/or 2009/02/01 and/or other schemes, and translate these into language-specific date strings.
- Perhaps study the 24-page electronic document *APA Style Guide to Electronic References*, see <http://books.apa.org/books.cfm?id=4210509>, and incorporate my findings in the next version of `apacite`.
- Rewrite and reorganize the documentation; perhaps create a “getting started” document.
- Transform the examples section below into a test document. (You can already extract its contents from the file `apacite.dtx` and copy it into a `.tex` file for this purpose, but you’ll have to edit a few things in and out.)
- Improve the source code documentation, esp. the formatting, indexing, and cross-referencing. (You only see this if you regenerate this manual with the `\AlsoImplementation` flag set.)

Quite likely, other problems will come up when using the current version of `apacite`, but at least I got it working on the APA manual examples, see section ?? below. I have lots of ideas for improvements, extensions, options, etc., but not much time to devote to it. Therefore, it is difficult to give a good estimate of when I will release a new update. However, in the meantime, if you have questions, remarks, suggestions, or bug reports, you can e-mail them to me.

## 13 Examples of the APA manual

In this section, the `apacite` package is tested by citing the examples of the APA manual (5th ed., pp. 207–281), and some additional references for additional purposes.

The following list gives the examples from chapter 3 of the APA manual, with section number. For these and all later examples, whenever I thought it necessary or desirable, I added some comments. These are indicated by the  $\triangleright$  symbol. Especially for later examples (from chap. 4), the comments frequently pertain to the corresponding reference list entry.

3.94 ? (?) compared reaction times

In a recent study of reaction times (?, ?)

In ? ? compared reaction times

In a recent study of reaction times, ? (?) described the method. . . ? also found

3.95 ? (?) found

? (?) found

? found

(?, ?) and (?, ?)

? (?) and ? (?)

$\triangleright$  It seems that if “et al.” would refer to one additional author, then this author is named instead of the “et al.” clause.

? (?) and ? (?)

as ? (?) demonstrated

$\triangleright$  The comma after “Nightlinger” in the reference list does not look right, but is explicitly required by the APA. Of course, with people as authors, one almost always has initials (although I have seen an example of an econometrician who only had one name), so this problem is not likely to occur. With nicknames (“screen names” as they are called in the

APA manual below ex. 85) in messages to newsgroups, initials are typically missing, but then there is usually (always?) only one author, so the problem does not exist either. The problem can, however, come up when there are two authors, the first of which is a corporate author, or with artists, e.g., the song “Into the Groove” is written by Madonna and Stephen Bray.

as has been shown (?, ?)

### 3.96 (?, ?)

(?, ?)

(?, ?)

(?, ?)

▷ If the **firstkey** field is different for different references and the **key** field is the same, then, analogous to the use of extra authors in 3.95 above, **apacite** uses the **firstkey** field for all citations:

? (?), ? (?), ? (?), ? (?);

? (?), ? (?), ? (?), ? (?).

Although this is a constructed example, these organizations really exist. The “Koninklijke Nederlandse Schaakbond” is the Royal Dutch Chess Association and the “Koninklijke Nederlandse Schaatsbond” is the Royal Dutch Skating Association. Both are abbreviated to KNSB and presumably, both publish an annual report each year. Furthermore, in these examples, I copied the contents of the **author** field to the **publisher** field. **apacite** recognizes this and puts the APA-requirement “Author” as publisher description in the reference list. Other styles may prefer the full publisher name again.

### 3.97 on free care (?, ?)

▷ The order of the closing quotes and the comma is different from the APA manual. I think this one is more logical, although the APA manual’s is the conventional one, presumably for aesthetic reasons. I hope to implement this in a future version.

the book ? (?)

(?, ?)

▷ If two articles both have no author and the title and the year are the same, a’s and b’s should be used: ? (?), ? (?), and ? (?). The latter two entries are correctly sorted by month, through the way the month macros are defined in the **.bst** files.

### 3.98 ? (?) and ? (?) also found

? (?) and ? (?) studied

### 3.99 Past research (?, ?, ?)

Past research (?, ?, ?, ?)

Several studies (?, ?, ?, ?, ?, ?, ?)

Several studies (?, ?, ?, ?)

▷ **apacite** does not (yet?) sort the cited references in the text (of course it does for the reference list), as required by the APA manual.

(?, ?; see also ?, ?, ?)

▷ Note that I had to use **\citeNP** here.

### 3.100 (?, ?)

▷ This is not an example, but a rule in the text. The “n.d.” is handled by the **\bibnodate** macro. It is not clear whether it is really meant that the in-text form is ?, ? or the more logically consistent ? (?). The former can (currently) only be accomplished by using



`\citeauthor` and `\citeyearNP` explicitly, whereas the latter is simply obtained by `\citeA`.  
(?, ?)

▷ Again, this is not an example, but a rule in the text.

(?, ?)

? (?)

▷ For citations to the bible, I used `\biblecite` and several obvious variations on it, as discussed in section ??:

`\bibleciteA`, first cite: 1 Cor. 13:1 (Revised Standard Version)

`\bibleciteA`, second cite: 1 Cor. 13:1

`\biblecite`, first cite: (1 Cor. 13:1, Revised Standard Version)

`\biblecite`, second cite: (1 Cor. 13:1)

`(\bibleciteNP`, first cite: 1 Cor. 13:1, Revised Standard Version and some text surrounding it)

`(\bibleciteNP`, second cite: 1 Cor. 13:1 and some text surrounding it)

▷ Here, there probably should be a comma if it's the first cite and not a comma when it's a later cite. A difficult one.

▷ Ancient Greek and Roman works should be treated similarly, i.e., not be listed in the reference list. But it seems to me that title and/or author must be given in the text as well. Maybe I will try to construct a tentative example of this in a future release.

3.101 (?, ?, p. 332)

(?, ?, chap. 3)

▷ If you like, you can use the `apacite`-defined abbreviation commands `\BPG` and `\BCHAP`.

(?, ?, ¶ 5)

(?, ?, Conclusion section, para. 1)

3.102 ▷ Personal communication is not really citation and should be done manually:

T. K. Lutes (personal communication, April 18, 2001)

(V.-G. Nguyen, personal communication, September 28, 1998)

3.103 (see Table 2 of ?, ?, for complete data)

Chapter 4 of the APA manual deals with the reference list and only mentions some in-text citation issues in passing. Therefore, the rest mainly requires a lot of moving back and forth between the reference list and this part. I start with the more general part, sections 4.01–4.15, presented similar to the sections from chapter 3 as presented above.

4.04 ▷ First, some examples mentioned in the text:

? (?), ? (?), ? (?), ? (?), ? (?), ? (?), ? (?), ? (?), ? (?)

▷ The APA manual states that prefixes must be treated according to the rules of the language of origin. These rules can differ a lot between languages (and countries) and it is impossible to know all these rules. The APA manual gives two examples, one in which “De Vries” is treated as the surname, and one in which “Helmholtz” is the surname and “von” is the “von” part of the name.

In the former case, the author is referred to as, say, “De Vries (1999)” in text and “De Vries, J. (1999).”, alphabetized under “D” in the reference list. In the second example, the author is referred to as, say, “Helmholtz (1870)” and listed in the reference list as “Helmholtz, H. L. F. von. (1870).”, alphabetized under “H”. To get this right, the author must be defined as

```
author = {H. L. F. von Helmholtz},
```

or

```
author = {von Helmholtz, H. L. F.},
```

then `BIBTEX` will assign the correct parts to the first names, “von”-part, and last name. The seemingly logical

```
author = {Helmholtz, H. L. F. von},
```

will not be formatted correctly, because “von” is now considered to be part of the first names and therefore abbreviated to “v.”.

Below, under example B-2, I will further discuss some issues about “von” parts.

▷ The APA manual requires that numerals are alphabetized as if they were spelled out. This is not done by `apacite`, so when applicable should be done by the user, e.g. by using the `\APACSortNoop` command as discussed in section ?? . I give one fictitious example (?, ?).

? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)  
? (?)

▷ As stated in the APA manual, an exception to the rule of sorting by title if author and year are the same occurs when the works are identified as a series. Then ordering the parts of the series sequentially can be done using the `\APACSortNoop` command.

▷ Here, it says that only “A” and “The” are excluded from the sorting (if they are the initial words), but I assume they mean “An” as well. This is how it is implemented in `apacite`.

? (?)  
? (?)

▷ I have not (yet?) studied referencing legal materials. In the first version of `apacite`, I defined a `literal` type, so that the users can literally format such entries themselves. This is still available. However, I think that if you need to refer to legal cases, you can find a way to use the `misc` type to get it right. I might study this and give examples in a next version of `apacite`.

4.05 ? (?)  
? (?)

▷ Here I used the `\nocitemeta` command to let `apacite` know that a meta-analysis is reported and this work is included in it. As you can see, you can still cite the work by using the normal citation commands such as `\citeA`.

4.08 ? (?)  
? (?)

? (?)

▷ To obtain the “(with ...)” description in the reference list, I used the `\biblital` command. Furthermore, to get this formatted correctly in the author index, I used the `\indexskipbracenodot` command to suppress an additional period. These commands are described in section ??.

To BibT<sub>E</sub>X, the `\biblital` construction is a complicated accented character, which is the second initial of the author. Consequently, this does not work properly with styles that put the initials *before* the surnames, and care must be exercised with “junior” parts (see ex. 70). Furthermore, this construction must be put between a pair of braces in order for BibT<sub>E</sub>X to treat it as an accented letter, instead of abbreviating it.

4.09 ? (?)

4.10 ? (?)

? (?)

4.11 ? (?)

? (?)

▷ Note that in the `.bib` file, the “junior” part comes directly after the surname and a comma, with the initials after the “junior” part, whereas in the formatted reference list, the more logical order  $\langle surname \rangle$ ,  $\langle initials \rangle$ ,  $\langle junior \rangle$ , is used.

4.12 ? (?)

4.13 ? (?)

▷ “For substantial reference works with a large editorial board, naming the lead editor followed by `et al.` is acceptable”. This is now implemented in `apacite` as follows: If there are 7 or more editors, only the first one is named, followed by “et al.” (?, ?).

4.15 ? (?)

▷ Here I used the new `url` and `lastchecked` fields.

? (?)

The rest is from section 4.16, the examples section. The section number is omitted, but A–I are used instead for the unnumbered examples. For the numbered examples, only the number is given. The following enumerated list gives the example number, and the two basic in-text citation commands, both in a full and short form, of the examples.

A-1 (?, ?)

(?, ?)

? (?)

? (?)

1. (?, ?)

(?, ?)

? (?)

? (?)

2. (?, ?)

(?, ?)

? (?)

? (?)

3. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

4. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

▷ Note that the list of authors can be finished with “**and others**” if there are more than 6 authors, or all authors can be given, in which case **apacite** truncates the list after the first 6. For compatibility with other styles, the latter is preferable.

5. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

6. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

7. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

▷ Here, I used

```
month = spring,
```

using the new **spring** macro defined in the **.bst** files.

8. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

▷ Here, again, the order of the closing quotes and the following comma should be reversed according to the APA rules.

▷ The **month** field is formatted as

```
month = aug # "/" # sep,
```

which uses the month macros and **BIB<sub>T</sub>E<sub>X</sub>**’s concatenation operator ‘**#**’.

9. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Here, again, the order of the closing quotes and the following comma should be reversed according to the APA rules.
10. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
11. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Apparently, a weekly newspaper is not a magazine and should therefore include “p.” or “pp.” in front of the page number(s), whereas a magazine article should not. I do not understand the distinction very well and find it not very useful as well, but will adhere to it. It means that the **newspaper** type gets the “pp.”, whereas **magazine** and **article**, which are now identical, do not.
12. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Here, `\emph{DSM-IV}` must be entered without an extra pair of braces in the title field; see ex. 73 for the intricacies of braces and `\emph` in the title field.
13. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Here, the **number** field is 1, **Serial No.**~231.
14. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Here, the **number** field is 3, **Pt.**~2.
15. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
16. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

17. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ This case is the opposite of the more common situation where a reprint is cited and an original publication year is given. I could implement similar code to handle this case (and I may if I get requests to do this), but for the moment, I decided to trick the system: I used the `\bibref{text}` command mentioned in section ?? . It must be called with two arguments. Initially, it places the second argument in the text, but just before the bibliography, its definition is changed so that the first argument is placed in the reference list. By defining

```
year = {{\protect\bibref{text}{1992}{1992/1993}}},
```

in the `.bib` file, the year is “1992” in the reference list and “1992/1993” in the text citations. The `\protect` is necessary to make sure that the `\bibref{text}` macro is not written out in the label-part of the bibliography, because then only “1992” would appear in the text reference. The additional pair of braces are necessary to make sure that the entry is alphabetized correctly. When these are omitted, the entry is alphabetized under the “P” of “protect”. Of course, the latter is only important if the reference list contains multiple works by the same author(s), but the same phenomenon is encountered when the `author` or `editor` field starts with a command, or when there is no author or editor and the `title` field starts with a command, as in the entry ? ( ? ) introduced in 4.04 above.

Of course, you can save yourself a lot of trouble by obtaining the original article and citing that.

18. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ Here, the `number` field is `Suppl.~2`.

19. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ Other styles treat the *Annual Review* as a book (*Sociological Methodology* is an even stronger example), so this is a point where the `.bib` file becomes highly incompatible between styles. This cannot be avoided.

20. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ Here, the `englishtitle` field is used for the English translation of the German title. To get the German capitalization of the title right, the capitalized words must be protected by an additional pair of braces.

21. (? , ?)  
 (? , ?)

? (?)

? (?)

▷ Apparently, a translator and original publication date do not have to be mentioned here.

22. Seidenberg and McClelland's study (as cited in ?, ?)

▷ The APA website (<http://www.apastyle.org/faqs.html>) mentions that the year should generally also be mentioned, making it a complete cite, except that there is no reference list entry. In my opinion, it would be better to list the complete original work in the reference list as well, in case readers do have access to the original work, but not to the secondary source. But this is not the APA rule.

▷ You may like to have the authors of the unlisted reference in the author index as well. Use the `\AX` command for this, as explained in section ??.

B-1 (?, ?)

(?, ?)

? (?)

? (?)

23. (?, ?)

(?, ?)

? (?)

? (?)

▷ Here, and in several other references, I put `\hbox{}` after the “Jr.”, so that `TeX` does not consider the period as a sentence-ending period.

▷ `apacite` now supports only an integer value (“3”) in the `edition` field, and transforms this to “3rd”.

24. (?, ?)

(?, ?)

? (?)

? (?)

▷ The APA manual gives a number, which would not ordinarily be done with proper books and which suggests that it could also be treated as a report. In that case, the default “Tech. Rep.” description should be left out, however. This can be accomplished by defining

```
type = {\bibnotype},
```

which flags that the type must be left out, see ex. 42. However, for books, it now works fine leaving the `type` field empty and giving the number in the `number` field.

25. (?, ?)

(?, ?)

? (?)

? (?)

▷ According to the note after ex. 25 in the APA manual, an edited book with “just one” author should list the editor (and translator, when available) after the title, just like the translator. An example of this is ? (?). (Should the original years of publication be used here?)

However, what does “just one author” mean here? Does it mean that if there are 2 authors and 1 or more editors, the editors should not be mentioned, or the authors should not be

mentioned? Presumably, it means the following: In the default case, an edited book consists of several chapters, with different chapters written by different authors. In this case, the reference to the whole book lists only the editor(s). If every chapter is written by the same author, and there are 1 or more editors, their condition holds. I think that if all chapters are written by the same  $n$  ( $\geq 2$ ) authors, it should also be treated in this way, i.e., with the authors in the author position and the editor behind the title. A difficult case would appear if the book were a collection of works of, say, 2 authors, with some chapters written by one and some by the other, and some jointly, as in ? (?). These choices have to be made by the user, however, **apacite** will format author, editor, and translator, whenever available.

26. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ According to section 3.97, the title should be in italics. In this example, in the citation, it is not given in italics, but this must be an error, because they do give it in italics on p. 219 (but give no date there).
27. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Here, the **edition** field has value “Rev.\hbox{” and **apacite** handles this noninteger value correctly as well.
28. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Use the en-dash (“--”) and not simply a hyphen (“-”), because the latter is not considered to mean “to” by **apacite** (cf. “Vol. A-1”).
29. ▷ As mentioned in section ??, **apacdoc.sty** provides the **\DSMcite** macro and some related alternatives to get the *DSM* references right. On rereading the description in the APA manual, I discovered that previous versions of **apacite** did not format the first citation correctly, although the description is a bit ambiguous as to how exactly the first citation should be formatted. Therefore, I looked at a recent issue of *Journal of Abnormal Psychology*, 116(3); and *Psychological Assessment*, 19(2). These show some slight variations, some of which have good reasons, but others seem arbitrary. In all articles I looked at, the first citation was an in-text citation, and the following form was the most frequently used, and thus implemented in **apacdoc.sty**:  
**\DSMciteA**, first citation: ? (?; ?; ? , ?)  
 ▷ The following citations then only use the abbreviated title:  
**\DSMciteA**, second citation: ?  
 ▷ Because I haven’t found a parenthetical first cite, I have to use some logic and common sense to define it:  
**\DSMcite**, first citation: ( ? , ?; ?; ? , ?)  
 ▷ Subsequent parenthetical cites are formatted as  
**\DSMcite**, second citation: ( ? )



▷ Finally, here are “NP” versions of them:

`\DSMciteNP`, first citation: ?, ?; ?; ?, ?

`\DSMciteNP`, second citation: ?

To use these citation commands, you have to provide a **key** field that contains a `\bibDSM` command:

```
key = {\bibDSM{American Psychiatric Association}{Diagnostic
and Statistical Manual of Mental Disorders}{4th
ed.\hbox{}}{DSM-IV}},
```

As is evident from this example, the four arguments are author, full title, edition, short title. Thus, because all this information is provided in the `.bib` entry (and not preprogrammed in `apacite`), it is conceivable that this command is useful for other types of works as well. The generality is a bit limited, though, because all four arguments must be available and have meaningful content (i.e., should not be empty). But it is relatively straightforward to write your own specialized citation commands, based on the `\DSMcite` code, which do not require all parts to be available.

However, the `\DSM*` macros are still very rough. Multiple citations are not handled well, so only one work should be cited with such a macro. Furthermore, the pre- and postfixes (through `<...>` and `[...]`) are not available. Note also that the edition should be spelled out as “4th ed.\hbox{”}. When I experimented with just “4”, trying to use the `\PrintOrdinal` function, things went terribly wrong. Even if I put `\PrintOrdinal{4}` in the **key** field, it didn’t work. Finally, the “ed.” must be added to allow reference to the “3rd ed., rev.” and the “4th ed., text rev.” as well.

▷ In some of the articles I studied, ordinary citations were used, esp. when page numbers were given in parenthetical cites. Thus, you can also use

`\cite[p.~25]{ex29}`: (?, ?, p. 25)

30. (?, ?)
- (?, ?)
- ? (?)
- ? (?)

▷ Here, “6th ed.” comes before “Vols. 1–20”, so it is Vols. 1–20 of the 6th edition. Pre-[2003/09/05] versions of `apacite` placed volumes before editions, which is appropriate when different volumes of a series are not updated jointly, so that you may have the 3rd edition of Vol. 2, but only the 1st edition of Vol. 6. Should you want this, you have to trick the system by putting both elements in the **edition** field:

```
edition = {\BVOLS\ 1--20, 6th},
```

and leaving the **volume** field empty.

▷ Does the remark about large editorial boards mean that the APA intended to add “et al.” here? Anyway, here is an example of such a situation: ? (?).

31. (?, ?)
- (?, ?)
- ? (?)
- ? (?)

32. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

33. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ This illustrates that a `type` field can be used with a `book` entry.

B-2 (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ This is a suitable point to discuss “von” parts further. One of the editors in the current example is called P. van den Broek. I am Dutch and I recognize this as a Dutch name, so let’s assume that this person is Dutch. Let us now consider the situation that this person is the (only) author. Then according to the Dutch rules, this person would be listed in the reference list as “Broek, P. van den. (1992)” and listed under “B”, but referred to in the text as “Van den Broek (1992)”, including the “von” part and capitalizing the first letter. When an initial precedes the “von” part (e.g., when there are two primary authors with this surname, but with different initials), then this person would be referred to in the text as “P. van den Broek (1992)”, including the “von” part, but now in lower case. The Dutch rules are very difficult to do right in `BIBTEX`/`LATEX` and they conflict with the rules for other languages, and presumably with the rules of the APA, which would require this person to be alphabetized under “V”, at least.

For the current editor position, there is no big problem with the APA rules. The “von” part can really be put in the “von” part as recognized by `BIBTEX`, which I did, or it can be enclosed in braces, making the whole last name one part. Either will come out right and the editor will not be referred to in the text anyway. With styles that put the initials behind the surname, it will only work right when put in the “von” part. In the author position, however, things are different. Then this author could be formatted as

```
author = {Van {\lowercase{D}}en Broek, P.},
```

where the `{\lowercase{D}}` construction is used to prevent `BIBTEX` from interpreting it as a “von” part, or

```
author = {Van{\ }den{\ }Broek, P.},  
author = {{Van den Broek}, P.},
```

which in most situations gives the correct version according to the Dutch rules with the correct alphabetizing according to the APA rules. Only in situations when the initial would appear *before* the surname (when there are other authors with the same surname or when using other styles), this would lead to unwanted results.

If you really want to alphabetize according to the Dutch rules (which is *not* APA in my interpretation), you can use the following trick. `apacdoc.sty` defines a macro `\Dutchvon`

with two arguments, which is equivalent to the `\bibref{text}` macro discussed in ex. 17 above. That is, in the beginning of the document, it is defined as echoing its second argument, and just before the reference list, it is redefined to echo its first argument. Then define the author as

```
author = {van den {\protect\Dutchvon{Broek}}{Van den Broek}}, P.},
```

which, combined with the `apacite` formatting implies that in text, “Van den Broek” is used, whereas the author is alphabetized under “B” in the reference list, and listed as “Broek, P. van den”. Of course, we could simply have used `\bibref{text}` again, but that was used to cover a completely different situation, and I can easily imagine that when switching to another citation style, the former application (i.e., 1992/1993) must be kept, but the “von” part is handled differently, so that `\Dutchvon` must be redefined. Based on this example, users should be able to define their own tricks for similar peculiarities.

Finally, note that in Belgium, where many people speak Dutch as well, it is customary to alphabetize under “V” anyway (“Van Damme”).

34. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
35. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ The series title, volume number, and volume title are all put into the `booktitle` field.
36. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Again, the series title, volume number, and volume title are all put into the `booktitle` field.  
 ▷ I had to use some tricks to get the Series and Volume Editors’ descriptions right. The “(Series Ed.)” is according to `BIBTEX` part of the last name of the first editor, and I used the `\bibeditortype` macro from `apacdoc.sty` to get the volume editor right. See section ?? for my interpretation of APA’s usage of the term “series”.
37. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Note that “Van” in the first editor’s surname is capitalized here, as in the APA manual. In the original Dutch, this would be in lower case.
38. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

39. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

▷ Here, it is stated that the translator's name should be put after the editor's name when both are different, but in ex. 40, the translators are put after the title, before the editors. Apparently, if the book is a collection of works by one author(-group), edited and translated, then the translator should come after the editor, but if different chapters are translated by different translators, or perhaps not all are translated works, then the translator should come after the title. Of course, **apacite** has no way of knowing this. In the current implementation, for an **incollection**, if the editor and translator are the same, they are formatted as in this example, whereas if they are different, they are treated as in ex. 40 below. I think this will be satisfactory in most cases. If you really want the translator after the editor when they are different, you can trick the system in a way similar to ex. 36.

40. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

▷ See my comments to ex. 39 above.

A test of the **originaljournal** formatting: ? (?).

- C-1 (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

▷ The APA manual initially formats the number with an en-dash: "91–5". When it discusses the number separately, however (p. 256), it uses a single hyphen: "91-5". Logically, the latter makes more sense to me (and is generally used in other numbers in the APA manual as well), so that's what I did here. Moreover, if the en-dash would be entered as "--" in the **number** field, **apacite** would interpret this as "to" and insert "Nos." instead of "No.". Thus, if you want the en-dash, you have to put this in the **.bib** file in a different way, e.g., using the **\textendash** command.

41. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

42. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

▷ I used the **\bibnotype** command to suppress the default "(Tech. Rep.)" description. Actually, I don't think there is anything wrong with calling a report a report, be it "technical" by default, or by specifying

```
type = {Report},
```

except that it's not done so in this specific example in the APA manual. But there does not seem to be a rule against it.

▷ The NTIS information is put in a **note** field.

43. (? , ?)  
(? , ?)  
? (? )  
? (? )

▷ The ERIC number is given in the **note** field.

44. (? , ?)  
(? , ?)  
? (? )  
? (? )

45. (? , ?)  
(? , ?)  
? (? )  
? (? )

46. (? , ?)  
(? , ?)  
? (? )  
? (? )

▷ Note that in the example in the APA manual, “Western Australia” *is* mentioned in the publisher location, whereas their rule explicitly states that this should not be the case in this situation, and they stress that again below the example. Therefore, I did not mention “Western Australia” in the **publisher** field.

47. (? , ?)  
(? , ?)  
? (? )  
? (? )

48. (? , ?)  
(? , ?)  
? (? )  
? (? )

49. (? , ?)  
(? , ?)  
? (? )  
? (? )

50. (? , ?)  
(? , ?)  
? (? )  
? (? )

51. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ As of this version of **apacite**, the location of a **lecture** can be given in the **address** field.
52. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
53. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
54. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
55. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ It seems logical to suppress the original year in the citation if it's the same as the year of the abstract, cf. the comment to ex. 17 in the APA manual (p. 245).  
 ▷ According to example 54 and the text below it, an "A" or "B" should be added to the volume number of *DAI* of this entry, presumably an "A", but because they did not give one, I left it out as well.
56. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
57. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ In ex. 59, the city should not be mentioned because it is already mentioned in the name of the university. Here, the city *is* mentioned again, which seems contradictory. This is all in the **.bib** file; **apacite** uses whatever is given there, so this is not a formatting problem for **apacite** itself.
58. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

59. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ See ex. 57 for the city issue.
60. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
61. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ The description of the data goes into the **type** field. It may be more logical to put it in the **title** field, with some trickery to indicate that it's a description and not a title, but putting it in the **type** field already ensures that it's formatted correctly, so I think I'm gonna be lazy on this one. The same applies more or less to reviews (G-1, 63, 64), although there it is logical to put at least "Review" in the **type** field, so as we're already there, we might as well make it complete.
62. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ "Available from" an address should be spelled out in the **note** field, to get the parentheses. "Available from" a website is without parentheses, see ex. 95.
- G-1 (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ See my comments to ex. 61 above about the logic of the **type** field. For a review, the **type** field must contain the message that it is a review of a *book* (or motion picture, or whatever it is a review of) and give the title of the reviewed work in italics, which must be manually formatted, e.g., by using the `\APACcitebtitle` macro. The author of the reviewed work should apparently not be mentioned.
63. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )
64. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 ▷ Here, the title of the motion picture, *Discovering Psychology*, is capitalized, whereas this was not done with book titles in ex. G-1 and 63. I don't know whether this is an error or an

implicit rule, but this formatting must be done in the `.bib` file, so it's up to the user to do this as desired.

65. ▷ The rules for movies, television series, etc. do not appeal very much to me, although they are not nearly as bad as those for music recordings (ex. 69 and further, see below). My primary objection is that when referring to movies and TV series, the title is the most important characteristic. The director may be a good second in some cases, but producers and writers are almost always unknown to the general public and not relevant when referring to them. You talk about the movie *Jaws* and not about Spielberg (1975) or whoever wrote the script. Recognizing the primary creative input or authorship may be politically correct, but not particularly informative. However, you may think that my objections are mainly driven by the difficulties of trying to format the entries in `BIBTEX`. Anyway, here are my attempts to implement the APA rules.

(?, ?)

(?, ?)

? (?)

? (?)

▷ I find “United States” a somewhat uninformative address.

(?, ?)

(?, ?)

? (?)

? (?)

▷ Note again that this and the following “Available from” must be between parentheses (accomplished by putting it in the `note` field), whereas “Available from” some web site must not be between parentheses, cf. ex. 95.

▷ In these examples, I used the `\bibliteral`, `\bibskipbracenodot`, and `\indexskipbracenodot` commands from `apacdoc.sty`.

(?, ?)

(?, ?)

? (?)

? (?)

▷ Here, I simply let “(Producer)” be part of the author name, but because of this addendum, the entry is not sorted correctly in the reference list: It should be before ? (?), but because of the addendum, the authors are different to `BIBTEX` and this one comes after ? (?). The ordering can be corrected by using the `\APACSortNoop` command as discussed in section ???. However, the authors of these entries will still be different. If it's important that they are the same (e.g., for the author index), the `.bbl` file should be edited manually. This seems an extremely rare event, though.

66. (?, ?)

(?, ?)

? (?)

? (?)

▷ Here, I used the same tricks as in ex. 65.

67. (?, ?)

(?, ?)

? (?)



? (?)

▷ I used several tricks introduced before: the `\bibeditortype` (cf. ex. 36) construction to be able to use “(Producer)” in the editor position instead of “(Ed.)”, and the ones used in the previous examples. Because there is another Miller as well, the initial is given here. To make sure that `BIBTEX` thinks that there is only one initial, but the “(Producer)” is formatted correctly, the name is defined as follows:

```
editor = {Miller, {\bibinitial{R\bibeditortype{Producer}}}},
```

using the `\bibinitial` and `\bibeditortype` commands from `apacdoc.sty`. It is then formatted correctly both in the text and in the reference list if `\bibeditortype` is defined to swallow its argument in the text, and redefined prior to the bibliography. After the bibliography (before the index), it has to be redefined again to swallow its argument.

68. (? , ?)

(? , ?)

? (?)

? (?)

▷ Again, I used several tricks that were also used in the previous examples.

69. ▷ The rules for music recordings do not make any sense, and they are not entirely clear as well. It would seem to me that most pop, rock, and other “light” music songs are almost always attributed to the performing artist. You can usually read who the composer and lyricist of the song are on the inlay-details of the CD, but if you want the reader to refer to “My Way”, the reader will easier find it (in a CD store or on the internet, for example) under “Frank Sinatra” than under the composer’s name (whoever that may be). Furthermore, the year of release of the CD (or other medium) seems more relevant in referring to the CD than the year of copyright of the specific song, or the recording date.

For classical music, things are different. Then, the writer (composer) is often more important than the performing artist, although the percentage of people who know who wrote the lyrics for Mozart’s operas may not be high. Furthermore, it may be time-consuming and not very relevant for most authors and readers to find the year in which a certain piece was written, although strictly speaking, we should use the “date of copyright”, which is not relevant for a lot of classical music.

However, when referring to Beethoven’s 9th symphony, say, it will typically not be very informative to refer to a specific recording on a specific CD, unless that specific recording is the subject of interest, but then it would seem that the “performing artist” (conductor or orchestra) should take the role of “author”. For example, when comparing a performance of this symphony conducted in 1975 by Masur with a performance of this symphony conducted in 1990 by Bernstein (assuming this exists), it does not seem logical to compare “9th Symphony” (Beethoven, 1823a, tracks 1–4) with “9th Symphony” (Beethoven, 1823b, tracks 1–4), but more logical to compare Masur (1975) with Bernstein (1990).

Finally, it seems a bit strange that the track number should be mentioned in text, but not in the reference list.

Again, you may think that my objections are mainly driven by the difficulties of trying to format the entries in `BIBTEX`. Nevertheless, here are my attempts to implement the APA rules, using variations of the `\citesong` command, defined in `apacdoc.sty`:

```

\citeAsong:  ?
\citesong:   ?
\citesongNP: ?
\citeAsong:  ?
\citesong:   ?
\citesongNP: ?

```

▷ The `key` field contains the relevant information in the form of a protected `\bibsong` command:

```
key = {\protect\bibsong{author}{year}{song title}{track info}},
```

Furthermore, I used some tricks to get the reference list entry right. I used the `incollection` type as a basis, with the CD title in the `booktitle` field. At first sight, it seems natural to define

```
type = {CD},
```

but with an `incollection`, the `type` field is put after the title, not after the book title. I may change this for music-types (CD, record, etc.), but this is currently not yet implemented. This means that the “[CD]” description must be put in the `booktitle` field, after the CD title and with explicit formatting commands:

```
booktitle = {Arkansas Traveler {\upshape{\uppercase{CD}}}},
```

On the other hand, we can now use the `type` field to put the recording artists in for the second example:

```
type = {Recorded by G. Bok, A. Mayo, \& E. Trickett},
```

We have to change the “In” string to “On” (“In” a book versus “On” a CD). This is done with a macro similar to the `\bibeditortype` macro discussed in ex. 36. In this case, the macro is `\bibInstring`, defined in `apacdoc.sty`, which has one argument. The macro changes the definition of the macro `\BIn` to the argument and then immediately changes the definition back to the old definition. So it makes a one-time change. Of course, the `\bibInstring` macro only works if it is executed *before* the `\BIn` macro. The latter macro comes before the `booktitle`, so `\bibInstring` must be added to a field that comes before that. I put it in the `title` field:

```
title = {Over the Waterfall{\bibInstring{\BOn}}},
```

where `\BOn` is defined by default as “On”. Finally, the recording date of the second example (1990) must be put in the `note` field.

If you want an author index, you would want the authors of these works in the index as well. Because of the usage of the `key` field, this is not automatically done for the citations. The `\AX` command has to be inserted explicitly, see section ?? above.

Note that this is all trickery within the  $\text{\LaTeX}$  domain and the `.bib` file, i.e., the user-definable and user-customizable area.

70. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

71. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )  
 (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

▷ In contrast with earlier versions of `apacite`, in which the “Retrieved ...” message had to be formatted in the `howpublished` field, this is now formatted automatically by using the `url` and `lastchecked` fields.

72. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

73. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

▷ Note that the APA manual breaks a line *after* a period instead of before, against their own rules. Furthermore, the APA manual uses a typeface in which ‘1’ (one) and ‘l’ (ell) are not distinguishable. This requires some guessing of the correct URLs. I think that this conflicts with their own stress on accuracy w.r.t. URLs.

▷ Note the subtle use of braces to get *Homo sapiens* correctly formatted: The capital in `\emph{Homo}` is retained by `BIBTEX`, whereas the capital in `{\emph{Sapiens}}` is changed into a lower case letter by `BIBTEX`, because of the extra pair of braces.

74. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

75. (? , ?)  
 (? , ?)  
 ? (? )  
 ? (? )

▷ Note that `BIBTEX` does not consider a question mark or an exclamation point as the end of the title and the beginning of the subtitle. Hence, if there is a subtitle after it, the capitalization of its first word is not retained (unlike after a colon), so we have to protect it using braces in the `.bib` file.

76. (? , ?)  
 (? , ?)

? (?)  
 ? (?)

77. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

78. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

▷ For this example, the source is not simply the URL, but with the qualification “Columbia University, Institute for Learning Technologies Web site:”. This means that we cannot simply use the `lastchecked` and `url` fields, but must revert to the way I proposed to format references to electronic sources in previous versions of `apacite`: by putting all the information in a `howpublished` field. This includes explicitly using the `\url` command and adding “`\bibnodot{.}`” immediately after the URL. The `\bibnodot` command, defined in `apacdoc.sty`, swallows the period, so that there is no period after the URL in the reference list, thus complying with the APA rules. This also applies to examples 79, 87, 91, and 95, and all the examples that do not list a specific URL, but only a description, like “the PsycARTICLES database” (ex. 88).

79. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

80. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

81. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

82. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

▷ Here and in ex. 83, I used the `\BAbstractRetrieved` and `\BmakeAbstractRetrieved` commands defined in `apacdoc.sty` to replace “Retrieved” by “Abstract retrieved”.

83. (? , ?)  
 (? , ?)  
 ? (?)  
 ? (?)

84. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

85. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ When referring to a message to a newsgroup, internet forum, etc., use the `misc` type. Then define

```
type = {\bibmessage},
```

and use the `number` field when relevant. The URL of the message is given in the `url` field. Then the entry is formatted correctly: If the message has a number, say 1, it reverts to “[`\bibmessage~1`]”, i.e., “[Msg 1]” with the default definition of `\bibmessage`. If there is no number, the type identifier is omitted (because it is mentioned as “Message posted to [...]” anyway). In both cases, the title is formatted as an article title, i.e., not italicized.

86. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

87. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

88. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

89. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

90. (? , ?)  
 (? , ?)  
 ? ( ? )  
 ? ( ? )

▷ Here, “de Ridder” may well be a Dutch name. If the APA manual is serious about its rule to format names as they should according to the rule of the country of origin, this name should then be formatted as “De Ridder”.

91. (? , ?)  
 (? , ?)

? (?)  
? (?)

92. (? , ?)  
(? , ?)  
? (?)  
? (?)

▷ From ex. 94 below, I deduce that the version information should be treated as an edition and not as part of the title. Therefore, in this example and ex. 94, the version is formatted in the `edition` field:

```
edition = {Version~4.0\noeedition\ignorespaces},
```

using the `\noedition` command from `apacdoc.sty`, which temporarily redefines `\BEd` to do nothing, and `\ignorespaces`, which eats up the space that is ordinarily present between the edition number and the “ed.” string.

93. (? , ?)  
(? , ?)  
? (?)  
? (?)

▷ To test the `manual` entry: ? (?).

94. (? , ?)  
(? , ?)  
? (?)  
? (?)

▷ The example in the APA manual does not list a date, not even a “n.d.”. Given the logic of the APA manual, this seems an error to me, so I added a “`\bibnodate`” in the `year` field, which becomes “n.d.” in the output. If it would really be the rule to omit the year, then it is not clear when a year must be omitted and when not, or replaced by “n.d.”. Furthermore, formatting in both the `.sty` file and the `.bst` file would have to be adapted to anticipate a missing year. Currently, I assume that there should always be something that acts like a date (a year, an “in press”, or a “n.d.”).

▷ See ex. 92 for the treatment of the version.

95. (? , ?)  
(? , ?)  
? (?)  
? (?)

▷ Note that here, unlike ex. 44 and ex. 91, “U.S.” is omitted before “Department of Health and Human Services”.