

notes2bib — Integrating notes into the bibliography *

Joseph Wright [†]

Released 2007/09/29

Abstract

The `notes2bib` package defines a new type of note, `bibnote`, which will always be added to the bibliography. The package allows footnotes and endnotes to be moved into the bibliography in the same way. The package can be used with `natbib` and `biblatex` as well as plain LaTeX citations. Both sorted and unsorted bibliography styles are supported.

Contents

	6	The mechanism	6
	6.1	The <code>endnotes</code> approach	6
	6.2	The REVTeX approach	6
	6.3	The <code>notes2bib</code> approach	6
1 Background	1		
2 Basic use	2	7 The package code	7
2.1 Package options	2	7.1 Package setup code . . .	7
2.2 Output of notes	3	7.2 User configuration macros	10
2.3 Cross-referencing notes .	3	7.3 User macros	11
2.4 Interaction with other		7.4 Internal setup macros . .	12
packages	3	7.5 Internal macros	13
3 Customisation	4	8 Change History	16
4 Special effects	4	9 Index	16
5 Known issues	5	10 Notes	17

1 Background

In most subject areas, bibliographic citations and notes are separate entities. However, in some parts of the physical sciences (chemistry and physics) it is usual for references to the literature and notes to be given together in a “References and Notes” section. By default, this requires that BibTeX users create a notes database for each document that they write.

The `endnotes` package allows the user to create endnotes rather than footnotes. However, this does not place the notes in the bibliography. The APS have

*This file describes version v1.1, last revised 2007/09/29.

[†]E-mail: joseph.wright@morningstar2.co.uk

developed the REVTeX document class, which allows footnotes and endnotes to be added to the bibliography. Notes can only be placed at the end of the bibliography using this system. Furthermore, the code to achieve this effect is not available as a package separate from REVTeX.

The aim of the `notes2bib` package is to make integration of notes into the bibliography easy. Notes can be written as normal in the LaTeX source, and are automatically moved to the bibliography. The package is compatible with sorted and unsorted bibliography styles. The package has been designed for use with numerical citations, although it will work with other systems.

2 Basic use

`\bibnote` In the most basic form, the package can be used simply by loading it in the preamble as normal. This adds a new type of note to the existing `\footnote` type: `\bibnote`. This can be used in exactly the same way as a footnote, taking one mandatory argument `{\langle text \rangle}`. The `\langle text \rangle` will be made available as to the bibliography as a note (henceforth referred to as a bibnote).

A very simple example of a bibliography note [1]. `\bibnote{Note for the first example}.`

By default, each bibnote is given an automatically-generated label (see Section 3 for details of modifying the automatic system). However, `\bibnote` accepts an optional argument `[\langle label \rangle]`, which can be used to over-ride this. This is particularly useful when a note will be referenced several times (The use of the `\citenote` command is covered in Section 2.3).

An example of a named note [2]. The text can then continue and reference the note again later [2]. `\bibnote[labelled-note]{Note for the second example}.`
The text can then continue and reference the note again later `\citenote{labelled-note}.`

`\bibnotemark` In common with `\footnote`, the basic `\bibnote` macro has companion macros `\bibnotemark` and `\bibnotetext`. The text provided for each not is not “fragile,” and so it should not be necessary to use `\bibnotemark` directly. `\bibnotetext` It is needed when replacing footnotes by bibnotes. Notice that there *are* places where bibnotes will be problematic, for example in section headings which also appear in the Table of Contents. In these contexts, use `\citenote` to reference the note, or use an optional argument to the `\section`, *etc*.

It is hard to write a good example for this [3]! The text continues here . It is hard to write a good example for this `\bibnotemark!`
The text continues here `\bibnotetext{Note for the third example}.`

2.1 Package options

The `notes2bib` package currently accepts a number of options. The `footnotes` option turns standard LaTeX footnotes into bibnotes.¹ Thus, with the `footnotes`

¹The `\thanks` macro used in making titles uses the footnote mechanism. This is unaffected by loading `notes2bib` even with the `footnote` option.

option enables, footnotes appear in the bibliography with no further user input. The `\footnotemark` and `\footnotetext` macros are supported, and are aliased to `\bibnotemark` and `\bibnotetext`, respectively.

The `endnotes` option works in the same way as the `footnotes` one, but in this case `\endnote` text is added to the bibliography. The `\endnote` command is defined by the `endnotes` package, but it is not necessary to load the `endnotes` package for this function to work in `notes2bib`.

The `head` and `tail` options affect the placement of notes relative to “real” citations. By default, all bibliographic citations are written in the order they appear in the source. Thus, with an unsorted bibliography style, they will appear in the order given. The `tail` option will force all notes to appear after real citations, whereas the `head` option causes notes to come first.² This has been designed to work with both sorted and unsorted bibliography styles.

2.2 Output of notes

Bibnotes are only printed when a bibliography is created. This means that at the very least a `\bibliographystyle` command must appear in the source. Under most circumstances, the user will be citing literature, and so will also include a `\bibliography` command in their source. Bibliography notes are automatically added to the citations to be printed.

`\printbibnotes`

If bibnotes are being used without any other citations, then the user cannot place `\bibliography` in the source.³ The package therefore provides the macro `\printbibnotes`, which will output only the notes. If the `endnotes` package has been loaded, the `\theendnotes` macro is redefined to achieve the same effect.

2.3 Cross-referencing notes

`\citenote`

As explained above, each note is automatically assigned a label, or the user can provide one as an optional argument to the note. In either case, notes may then be cross-referenced. Notes are available to be cited directly using the `\cite` command. However, this can cause problems when using the `tail` option. The `\citenote` command is therefore provided. This is aware of the options, and will act correctly in all circumstances.

Cross-references to the note labelled earlier using `[2]` and using `[2]`.

Cross-references to the note labelled earlier using `\cite{labelled-note}` and using `\citenote{labelled-note}`.

2.4 Interaction with other packages

`notes2bib` is designed to work well with as many other packages as possible. It tries to avoid changing anything which other packages may rely on. That said, it is best to load `notes2bib` *after* other packages that affect citations, footnotes or endnotes. As usual, the `notes2bib` package should be loaded before `hyperref`.⁴

²The `head` option is provided primarily for consistency with the `tail` option. The system used to give “head” notes is something of a hack, and is *not* a core part of the package.

³LaTeX will complain if the user puts `\bibliography{}`.

⁴It is usually the case that `hyperref` should be the very last package loaded in the preamble.

The `notes2bib` package is compatible with the current release of `biblatex` (v0.6). As `biblatex` works very differently from other citation systems, it must be loaded *before* `notes2bib`. This allows the package to adjust correctly for some important differences in operation of the bibliography environment.

3 Customisation

The `notes2bib` package provides a number of possibilities for customisation. Some of these will be useful only in somewhat unusual circumstances. These more low-level macros all begin `\niib...`

`\bibnotename` Each note created by `notes2bib` is automatically given a label, unless one
`\thebibnote` has been specified by user as an optional argument. By default, this is the text
of `\bibnotename` (normally `Bibnote`), followed by the note number (counted
automatically). By redefining `\bibnotename` all notes can be given a new label,
still bearing the note number. More complex effects can be achieved by altering
`\thebibnote`. This is defined as `\bibnotename\the\value{bibnote}` by
default.

`\niibcite` The macro used to cite notes is set in `\niibcite`. By default this is `\cite`, but
can be altered, for example `\renewcommand{\niibcite}{\citep}` if using
`natbib`.

`\niibprefix` The temporary BibTeX database used to store the notes has the default
name `niib-\jobname.bib`. The prefix (“`niib-`”) can be altered by redefining
`\niibprefix`.

`\niibrecord` When the notes database is written, each record is saved as type `Misc`, and
`\niibfield` the text is stored in a `note` field. Both of these can be altered by redefining
`\niibrecord` and `\niibfield`. This could be useful with a bibliography style
that has an explicit “Notes” record type, for example.⁵

`notes2bib` attempts to maintain the correct ordering of notes with sorted
bibliography styles. This is done by setting up suitable information in the tempo-
rary database to ensure correct sorting. The mechanism for this is available for
modification.

`\niibkeyhead` For users of standard bibliography styles, this is achieved by generating a
`\niibkeynone` key field in the database, containing the label used for the note. When using the
`\niibkeytail` head or tail option, this is prefixed by `aaa` or `zzz`, respectively. The prefix can
be altered by redefining `\niibkeyhead` or `\niibkeytail`, for example when
numerical labels need to be taken into account. The package also provides a
`\niibkeynone` macro, which defaults to an empty prefix.

`\niibpresorthhead` When using `biblatex`, the `presort` field can be used to control sorting of
`\niibpresortnone` the citations. Again, three macros `\niibpresorthhead`, `\niibpresortnone`
`\niibpresorttail` and `\niibpresorttail` are provided, which are used in the correct context.
The default values of `ml`, `mm` and `mn`, respectively, are designed to achieve the
desired effect while allowing the user full control of sorting.

4 Special effects

`\flushcitestack` When using the head or tail options, citations are added to a stack as they are

⁵The `rscc` and `achemso` packages contain styles of this type.

made. This stack is then flushed to the `.aux` file at the end of the document. If references are given by chapter (or other unit), this may not give the desired effect. The `\flushcitestack` macro will cause all saved citations to be written at that point, and will reset the stack for continued use. This can therefore be used to control when citation occurs.

`\delaycite`

When using the `head` option, normal citations have to be saved to the stack described above. This requires the modification of the citation commands. By default, `notes2bib` modifies the `\cite` command and most of the standard citation commands of `natbib` and `biblatex`.⁶ However, if problems are encountered with citation order, the user can use the `\delaycite` macro to patch any arbitrary citation command. This macro takes a single argument `{\macro-name}`, which is the name of the citation command to modify *minus* the leading backslash. For example, the `\cite` command is patched using `\delaycite{cite}`. The `\delaycite` macro checks that a citation command is not already modified, and so may be used again on the same command with no ill effects.

If a sorted bibliography style is in use, and more than nine notes are created, the sort order will be incorrect. This is because by default `notes2bib` does not pad the automatically-created labels with zeros. To get the correct sort order, `\thebibnote` should be redefined:

```
\renewcommand*{\thebibnote}{%
  \bibnotename%
  \ifnum\value{bibnote} < 9 0\fi%
  \the\value{bibnote}%
}
```

5 Known issues

From v1.1, the method for writing notes to the BibTeX database has been modified. This means that bibnotes cannot contain verbatim text.⁷ This is the same as for normal footnotes, and so the usual work-arounds are applicable.

The next note contains some awkward text [4].

The next note contains some awkward text

`\bibnote{Some \texttt{\textbackslash verb}-like output}.`

Currently, when using the `head` option citations can take only one optional argument. This is fine for normal citation styles, but means that `biblatex` users may have some issues. The package author will look again at this if it is a serious problem. A work around is to cite as follows (here assuming you wish to use the `\cite` command):

```
\makeatletter
\begingroup
  \@filesfalse
  \niib@orig@cite[argument-one][argument-two]{citation}
\endgroup
\edef\niib@stack{\niib@stack,citation}
\makeatother
```

⁶Notice that `\nocite` is not modified

⁷Actually, they can, but the spacing will go wrong. LaTeX will only complain if a note ends with verbatim text. However, verbatim text is not supported in bibnotes: don't do it!

Where “argument-one,” “argument-two” and “citation” should be altered to the desired information.

The `cite` and `biblatex` packages move superscript citations after punctuation. When these packages are used with standard (not `head` or `tail`) mode of `notes2bib`, the same will occur for bibnote numbers. Unfortunately, the `head` and `tail` options prevent this “auto-magic” operation on *bibnote* numbers.

6 The mechanism

The mechanism for positioning notes in the bibliography is somewhat involved. Rather than expect interested users to read all of the code that follows, a condensed overview is given here. The thinking behind the system used is explained first, by considering the `endnotes` package and REVTeX class. Both of these provided inspiration for this package.

6.1 The `endnotes` approach

The `endnotes` package⁸ allows the user to generate endnotes in the same way as footnotes. In `endnotes`, the text of the note is written to a `.ent` file. This is achieved in an unexpanded form using the `\meaning` TeX primitive. To produce the list of endnotes, this file is read back into LaTeX, with the extra information `\meaning` also writes being stripped off in the process.

This method is relatively simple in concept, but obviously does not integrate with BibTeX. The use of `\meaning` for unexpanded output also means that information requires further processing before it can be included in the bibliography.

6.2 The REVTeX approach

REVTeX takes a similar approach to creating endnotes, but also allows footnotes to be converted into endnotes. This results in a file containing all of the non-literature citations in a document in a single external file (in this case a `.end` file). REVTeX also uses a different method to achieve unexpanded output, meaning that several macros are not “active” in notes.

The second part of the REVTeX approach is to (optionally) read the notes back into the document. This is achieved by modifying the `\bibliography` environment to output each note in the bibliography. This takes place *en masse*, after the normal citations.

REVTeX makes a number of modifications to LaTeX, and is dependent on using `natbib`. The method used is also not compatible with interspersing normal citations and notes.

6.3 The `notes2bib` approach

In `notes2bib`, notes are again written to an external file. However, in contrast to the methods those outlined above, `notes2bib` writes its output in the well-known BibTeX database format. All of the note text is written almost completely

⁸ <http://tug.ctan.org/macros/latex/contrib/misc/endnotes.sty>

unexpanded to the file, the only requirement being that the braces match within the argument.⁹

Each note results in a citation being placed by `notes2bib` in the `.aux` file. The `\bibliography` command is also modified so that the new database will be used by BibTeX. After the BibTeX run, the note text will appear in the `.bbl` file, in the same way as any other citation. Using an unsorted BibTeX style, this results in notes interspaced with the normal citations. For sorted styles, `notes2bib` allows various methods for controlling the placement of notes, based on writing appropriate fields in the BibTeX database.

The `head` and `tail` options use an additional technique. Both delay writing of selected citations (either notes or real ones) into the `.aux` file, to control citation position in unsorted bibliographies.

7 The package code

7.1 Package setup code

The package starts with the usual identification code.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{notes2bib}
3 [2007/09/29 v1.1 Integrating notes into the bibliography]
```

The package requires e-TeX, so before going any further, this is tested. This code is taken more-or-less verbatim from `biblatex`.

```
4 \@ifundefined{eTeXversion}
5   {\PackageError{notes2bib}
6     {Not running under e-TeX}
7     {This package requires e-TeX. Try compiling the document
8       with\MessageBreak 'elatex' instead of 'latex'. When using
9       pdfTeX, try 'pdfelatex'\MessageBreak instead of 'pdflatex'}}%
10  \endinput}
11 {}
```

With that check out of the way, Philip Lehmann's `elatex` package is loaded to extend various LaTeX commands. Early versions of `elatex` uses slightly different macro names, so a version check is needed.

```
12 \RequirePackage{elatex}[2007/07/13]
```

Option processing takes place. Redefining `\footnote` or `\endnote` is delayed until the start of the document, to avoid the danger of other packages being loaded and overwriting this.

```
13 \newif\ifniib@fn\niib@fnfalse
14 \DeclareOption{footnotes}{%
15   \niib@fntrue
16   \AtBeginDocument{%
17     \let\footnote\bibnote%
18     \PackageInfo{notes2bib}%
19       {Converting footnotes to bibnotes}%
20   }}
21 \DeclareOption{endnotes}{%
22   \AtBeginDocument{%
```

⁹Writing to the file uses the e-TeX `\unexpanded` primitive.

```

23 \let\endnote\bibnote%
24 \let\endnotemark\bibnotemark%
25 \let\endnotetext\bibnotetext%

```

If the user has used the `endnotes` package, they may still have `\theendnotes` commands in their source. So this is defined to print out all of the bibnotes.

```

26 \@ifpackageloaded{endnotes}%
27   {\let\theendnotes\printbibnotes}{}%
28 \PackageInfo{notes2bib}%
29   {Converting endnotes to bibnotes}%
30 }

```

Stephen Dalton suggested an option for all notes to come after “real” citations. This is done here, along with a mirror options to have all for the citations come first.

```

31 \newif\ifniib@tail\niib@tailfalse
32 \newif\ifniib@head\niib@headfalse
33 \DeclareOption{tail}{\niib@tailtrue}
34 \DeclareOption{head}{\niib@headtrue}

```

Finally, the options are put into effect.

```

35 \ProcessOptions

```

`\niib@fnmark` If the footnote option is active, there is some work to do because of the `\thanks` macro. This is all done using the `elateX` package. Firstly, it is necessary to back up the original definitions for `\footnotetext`, `\footnotetext` and `\thanks`.

```

\niib@fnmark
\niib@fntext
\niib@thanks
36 \ifniib@fn
37   \let\niib@fnmark\footnotemark
38   \let\niib@fntext\footnotetext
39   \let\niib@thanks\thanks

```

`\thanks` Now the `\thanks` command is patched to use the saved definitions. The patching is nested so that everything has to go right before things are finalised.

```

40 \patchcmd{\thanks}{\footnotemark}{\niib@fnmark}{%
41   \patchcmd{\thanks}{\footnotetext}{\niib@fntext}{%
42     \AtBeginDocument{%
43       \let\footnotemark\bibnotemark
44       \let\footnotetext\bibnotetext
45     }%
46   \PackageInfo{notes2bib}%
47   {Modifying \protect\thanks\space macro}%
48   }{%

```

If the following is executed, then things have gone wrong in some way. The `\patchcmd` macro shouldn’t mess up the definition of `\thanks`, but to be on the safe side the original version is restored here.

```

49   \let\thanks\niib@thanks%
50   \PackageWarning{notes2bib}%
51     {Could not patch \protect\thanks}}%
52 }{%
53   \let\thanks\niib@thanks%
54   \PackageWarning{notes2bib}%
55     {Could not patch \protect\thanks}}%
56 \fi

```


The `head` and `tail` options are mutually exclusive, so a check is made that both have not been selected. If they have, the package ignores both options after a suitable warning.

```

57 \ifniib@head
58   \ifniib@tail
59     \PackageWarning{notes2bib}
60       {You've given both ``head'' and ``tail'' as
61         options\MessageBreak I'm ignoring both!}
62     \niib@headfalse\niib@tailfalse
63   \else
64     \PackageInfo{notes2bib}%
65       {bibnotes will ``head'' other citations}
66   \fi
67 \else
68   \PackageInfo{notes2bib}%
69     {bibnotes will ``tail'' other citations}
70 \fi

```

For the `head` option, the normal citation commands are modified to get the necessary delay in writing to the `.aux` file.

```

71 \AtBeginDocument{%
72   \ifniib@head

```

The real `\delaycite` is activated. The `\cite` command is always defined, so it can be redefined in all cases.

```

73   \let\delaycite\niib@delaycite
74   \delaycite{cite}%

```

If `natbib` is loaded, there are several citation commands that will definitely be defined. They are all run through `\delaycite`.

```

75   \@ifpackageloaded{natbib}{%
76     \delaycite{citet}%
77     \delaycite{citep}%
78     \delaycite{citenum}%
79     \delaycite{citealt}%
80     \delaycite{citealp}%
81     \delaycite{citetext}%
82     \delaycite{citeauthor}%
83     \delaycite{citeyear}%
84     \delaycite{citeyearpar}%
85     \delaycite{Citet}%
86     \delaycite{Citep}%
87     \delaycite{Citealt}%
88     \delaycite{Citealp}%
89     \delaycite{Citeauthor}%
90   }{}%

```

For `biblatex`, some citation commands are always defined.

```

91   \@ifpackageloaded{biblatex}{%
92     \delaycite{Cite}%
93     \delaycite{parencite}%
94     \delaycite{Parencite}%
95     \delaycite{footcite}%
96     \delaycite{Footcite}%
97     \delaycite{textcite}%

```

```
98      \delaycite{Textcite}%
```

Now there are some citation commands that may be defined or not.

```
99      \@ifundefined{autocite}{}%
100      {\delaycite{autocite}}%
101      \@ifundefined{supercite}{}%
102      {\delaycite{supercite}}%
103      {}{}%
104      \fi%
105 }
```

7.2 User configuration macros

The `notes2bib` package makes quite a number of configuration defaults available as macros. These are all set up here. Most of these are of the form `\niib...`, as in the main they are unlikely to be altered by most users.

<code>\bibnotename</code>	By default, all of the new citations take the prefix “Bibnote.” The default can be altered, and this is set up here.
	106 <code>\newcommand*\bibnotename{Bibnote}</code>
<code>\thebibnote</code>	A counter is needed for the notes created. In analogy to other counters in LaTeX, this is given a <code>\the...</code> name. The user should not really need to use this macro, but convention dictate that it has a user-space name.
	107 <code>\newcounter{bibnote}</code> 108 <code>\renewcommand*\thebibnote{\bibnotename\the\value{bibnote}}</code>
<code>\niibprefix</code>	The package works by creating a new BibTeX database file containing the notes. By default, this file is called <code>niib-<code>\jobname</code>.bib</code> , but the prefix may be altered by redefining <code>\niibprefix</code> . The default follows the pattern used by the <code>biblatex</code> package (which uses <code>blx-</code>).
	109 <code>\newcommand*\niibprefix{niib-}</code>
<code>\niibrecord</code> <code>\niibfield</code>	When creating the temporary database of notes, the BibTeX records created are of record type <code>\niibrecord</code> , and the notes are stored in the <code>\niibfield</code> field type. For some BibTeX styles, a better choice than the defaults (<code>Misc</code> and <code>note</code> , respectively) may be available. Thus the values used are definable.
	110 <code>\newcommand*\niibrecord{Misc}</code> 111 <code>\newcommand*\niibfield{note}</code>
<code>\niibpresorthhead</code> <code>\niibpresortnone</code> <code>\niibpresorttail</code>	When using <code>biblatex</code> , sorting can be controlled using the <code>presort</code> field. Default values to be used are given here; these are “soft” as <code>biblatex</code> allows much more definite pushing to the top and bottom of the sorted list.
	112 <code>\newcommand*\niibpresorthhead{ml}</code> 113 <code>\newcommand*\niibpresortnone{mm}</code> 114 <code>\newcommand*\niibpresorttail{mn}</code>
<code>\niibkeyhead</code> <code>\niibkeynone</code> <code>\niibkeytail</code>	For traditional BibTeX styles, the <code>key</code> field can be used to influence the sort order. Default values to alter sorting are given here, but may need to be altered for other languages and so on.
	115 <code>\newcommand*\niibkeyhead{aaa}</code> 116 <code>\newcommand*\niibkeynone{}</code> 117 <code>\newcommand*\niibkeytail{zzz}</code>

`\niibcite` Bibliography notes are marked by citations, like any other bibliography entry. The citation command use is controlled here.

```
118 \newcommand*\niibcite{\cite}
```

7.3 User macros

`\bibnote` Each new `\bibnote` increments the note counter, then checks for an optional label, before handing off to the internal macro `\niib@bibnote`.

```
119 \newrobustcmd*\bibnote{%
120   \stepcounter{bibnote}%
121   \@ifnextchar[%
122     {\niib@bibnote}%
123     {\niib@bibnote[\thebibnote]}%
124 }
```

`\bibnotemark` The `\bibnotemark` macro works in the same way as `\bibnote`, but calls `\niib@mark` rather than `\niib@bibnote`.

```
125 \newrobustcmd*\bibnotemark{%
126   \stepcounter{bibnote}%
127   \@ifnextchar[%
128     {\niib@mark}%
129     {\niib@mark[\thebibnote]}%
130 }
```

`\bibnotetext` The text companion to the mark macro above, with no increment of the counter.

```
131 \newrobustcmd*\bibnotetext[1][\thebibnote]{%
132   \niib@text{#1}%
133 }
```

`\printbibnotes` To allow for the possibility of there being no other notes, a command to print only notes is given. In the `biblatex` case, the best that can be done is to issue `\printbibliography`.

```
134 \@ifpackageloaded{biblatex}{%
135   \let\printbibnotes\printbibliography%
136 }{%
137   \newrobustcmd*\printbibnotes{\niib@bib%
138     {\niibprefix\jobname}}%
139 }
```

`\flushcitestack` In order to delay citations to the end of the bibliography (and thus force others to the start), a “stack” is created of citations which need to be written to the `.aux` file. This is done here, and the stack is cleared so collection can begin again.

```
140 \newrobustcmd*\flushcitestack{%
141   \ifx\niib@stack\relax\else%
142     \expandafter\nocite\expandafter{\niib@stack}%
143     \global\let\niib@stack\relax%
144   \fi%
145 }
```

`\delaycite`

```
146 \newrobustcmd*\delaycite[1]{%
147   \PackageWarning{notes2bib}{%

```

```

148     \@nameuse{delaycite}can only be used in the document
149     body\MessageBreak Not modifying citation macro \@nameuse{#1}}
150 }

\citenote A “mode aware” command for referencing notes is needed (problems arise with
\cite and the tail option. Rather than overload \cite with all of the problems
that can bring, a new command is provided that can be guaranteed to work.

151 \ifniib@tail
152   \newrobustcmd*\citenote[2][\relax]{%
153     \begingroup%
154       \@fileswfalse%
155       \relax is used as a marker for an empty first argument.
156       \ifx#1\relax%
157         \niibcite{#2}%
158       \else
159         \niibcite[#1]{#2}%
160       \fi
161     }%

If tail is not active, then \citenote is simply an alias for the citation command
used by notes2bib.

162 \else
163   \newrobustcmd*\citenote{\niibcite}%
164 \fi

```

7.4 Internal setup macros

```

\niib@keyname If biblatex is in use, the key field in the BibTeX database should be called
“keysort,” whereas otherwise it should be “key.”

165 \ifpackageloaded{biblatex}
166   {\def\niib@keyname{keysort}}
167   {\def\niib@keyname{key}}

\niib@presort \niib@key The values taken by \niib@presort and \niib@key depend on the desired
positioning of notes in the bibliography.

168 \ifniib@head
169   \def\niib@presort{\niibpresorthead}
170   \def\niib@key{\niibkeyhead}
171 \else
172   \ifniib@tail
173     \def\niib@presort{\niibpresorttail}
174     \def\niib@key{\niibkeytail}
175   \else
176     \def\niib@presort{\niibpresortnone}
177     \def\niib@key{\niibkeynone}
178   \fi
179 \fi

\niib@msg To inform the user, the automatically-created BibTeX database needs to carry
suitable information on its source.

180 \edef\niib@msg{%

```

```

181 This is an auxiliary file used by the 'notes2bib' package.^^J%
182 This file may safely be deleted. It will be recreated as
183 required.^^J}

```

\niib@stack This macro is needed to store any citations at the end of the bibliography. Initially, this expands to \relax.

```
184 \let\niib@stack\relax
```

A switch is needed to tell if a note is being cited when using “head” mode.

```
185 \newif\ifniib@cn\niib@cnfalse
```

7.5 Internal macros

\niib@delaycite Normal citations have to be stacked up in the stack when in “head” mode. To do this, the citation command has to be modified to do this, while leaving note citations alone. The trick used in this is to set \@filesw to false while the citation takes place.

```

186 \newrobustcmd*\niib@delaycite[1]{%
187   \@ifundefined{#1}{%
188     \PackageWarning{notes2bib}
189       {Citation macro \@nameuse{#1}undefined\MessageBreak
190       \protect\delaycite\space cannot operate on this macro}
191   }{%
192     \@ifundefined{niib@orig@#1}{%
193       \expandafter\expandafter\expandafter\let\expandafter%
194       \csname niib@orig@#1\expandafter\endcsname\csname #1\endcsname%
195       \expandafter\renewcommand\csname #1\endcsname[2][\relax]{%

```

If a note is being cited, then there will be no optional argument, and ##1 will expand to \relax.

```

196       \ifniib@cn%
197         \csname niib@orig@#1\endcsname{##2}%
198       \else%
199         \begingroup%
200         \@fileswfalse%

```

If an optional argument has been given, then ##1 will expand to something other than \relax. So the original citation command is called with the optional argument.

```

201         \ifx##1\relax%
202           \csname niib@orig@#1\endcsname{##2}%
203         \else%
204           \csname niib@orig@#1\endcsname[##1]{##2}%
205         \fi%
206       \endgroup%
207       \edef\niib@temp{##2}%
208       \ifx\niib@stack\relax%
209         \xdef\niib@stack{\niib@temp}%
210       \else%
211         \xdef\niib@stack{\niib@stack,\niib@temp}%
212       \fi%
213     \fi%
214   }%

```

```

215     \PackageInfo{notes2bib}%
216     {Modifying citation type \@nameuse{#1}}%
217   }{%
218     \PackageInfo{notes2bib}%
219     {Already delayed \@nameuse{#1}}%
220   }%
221 }
222 }

\niib@bibnote Two steps are needed here, writing the text of the note to file (handled by
\niib@text, and marking the citation (using \niib@cite).
223 \long\def\niib@bibnote[#1]#2{%
224   \niib@text{#1}{#2}%
225   \niib@cite{#1}%
226 }

\niib@mark If only a mark is asked for, this is simply converted into a citation. This cannot be
done directly as the optional has to be processed correctly.
227 \def\niib@mark[#1]{\niib@cite{#1}}

\niib@cite The internal macro used to cite notes depends on the package options in force.
First, for head mode, the modified real citation commands need to know that
a note is being cited. So the Boolean \niib@cn has to be set correctly before
citation.
228 \ifniib@head%
229   \def\niib@cite#1{%
230     \niib@cntrue%
231     \niibcite{#1}%
232     \niib@cnfalse%
233   }%

For tail mode, the citation has to be added to the stack for use later. The citation
is then made with writing to the .aux file disabled.
234 \else%
235   \ifniib@tail%
236     \def\niib@cite#1{%
237       \ifx\niib@stack\relax%
238         \xdef\niib@stack{#1}%
239       \else%
240         \xdef\niib@stack{\niib@stack,#1}%
241       \fi%
242       \begingroup
243         \@fileswfalse%
244         \niibcite{#1}%
245       \endgroup%
246     }%

Without the head or tail options, none of these tricks are needed, and
\niib@cite is simply an alias for \niibcite. That in turn may well just be an
alias for \cite!
247   \else%
248     \def\niib@cite#1{\niibcite{#1}}%
249   \fi%
250 \fi%

```

\niib@text The “business end” of writing the notes to file.

```
251 \long\def\niib@text#1#2{%
```

\niib@out If this is the first note, then a new output stream is needed, otherwise it will
\niib@stream already be open.

```
252 \@ifundefined{niib@out}{%
```

```
253 \if@files%w%
```

```
254 \newwrite\niib@out%
```

```
255 \gdef\niib@stream{\niibprefix\jobname.bib}%
```

```
256 \immediate\openout\niib@out\niib@stream\relax%
```

The new file starts with the message that it has been automatically generated by notes2bib.

```
257 \immediate\write\niib@out{\niib@msg}%
```

```
258 \fi}%
```

```
259 {}%
```

The new record is now written to file. The \unexpanded e-TeX primitive is used to avoid expansion of macros in the note text. The only issue with this is the addition of spaces after command names; this is the reason verbatim text cannot be used in bibnotes.

```
260 \if@files%w%
```

```
261 \immediate\write\niib@out{%
```

```
262 @\niibrecord\string{#1,^^J%
```

```
263 presort = \string{\niib@presort\string},^^J%
```

```
264 \niib@keyname\space= \string{\niib@key#1\string},^^J%
```

```
265 \niibfield\space= \string{\unexpanded{#2}\string}^^J%
```

```
266 \string}^^J}%
```

```
267 \fi%
```

```
268 }
```

The \bibliography macro is patched to ensure that when it is executed the note file is also processed. biblatex does things very differently, but this actually makes it much easier to patch for.

```
269 \@ifpackageloaded{biblatex}{%
```

```
270 \expandafter\bib@gadd\expandafter\blx@bibfiles\expandafter%
```

```
271 {\niibprefix\jobname}%
```

```
272 }{%
```

\niib@bib The bibliography command is patched so that it will run on the automatically-
\bibliography generated BibTeX database.

```
273 \let\niib@bib\bibliography%
```

```
274 \renewcommand{\bibliography}[1]{%
```

```
275 \ifnum\the\value{bibnote} > \z@%
```

```
276 \niib@bib{\niibprefix\jobname,#1}%
```

```
277 \else%
```

```
278 \niib@bib{#1}%
```

```
279 \fi}}%
```

At the end of the document, any delayed citations are written to the .aux file, and the database file is closed cleanly.

```
280 \AtEndDocument{%
```

```
281 \flushcitestack%
```

```

282 \ifundefined{niib@out}{}%
283 {\immediate\closeout\niib@out}%
284 }

```

8 Change History

v1.0		\@ifnextchar for detecting optional argument	11
General: Initial public release	1		
v1.0a		\citenote: Macro made robust	12
\citenote: New macro	12	\delaycite: Initial definition prevents use in preamble	11
\niib@delaycite: Fixed a flaw with checking for existing redefinition	13	\flushcitestack: Macro made robust	11
v1.1		\niib@cite: Definition made option-specific	14
General: \Percent macro removed	1	Name change from \niib@citenote	14
Documentation improvements	1	\niib@delaycite: Added a sanity check on argument	13
Improvements to documentation and dtx file	1	Improved logging of function	13
License changed from GPL to LPPL	1	Macro made robust	13
Removed use of xspace	7	Made internal to prevent execution in preamble	13
Require e-TeX extensions	7	\niib@text: Combined \niib@text and \niib@@text	14
Several code sections re-factored	1	Improved awareness of files switch	14
\bibnote: Macro made robust	11	\printbibnotes: Macro made robust	11
\bibnotemark: Macro made robust	11		
\bibnotetext: Macro made robust	11		
Use \newrobustcmd rather than			

9 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

B	73, 74, 76–89, 92–98, 100, 102, <u>146</u> , 190	I	\ifniib@cn . . . 185, 196
\bibliography . . . <u>273</u>		\ifniib@fn 13, 36	
\bibnote . . <u>2</u> , 17, 23, <u>119</u>		\ifniib@head 32, 57, 72, 168, 228	
\bibnotemark <u>2</u> , 24, 43, <u>125</u>	E	\ifniib@tail 31, 58, 151, 172, 235	
\bibnotename <u>4</u> , <u>106</u> , 108	\endgroup . 160, 206, 245		
\bibnotetext <u>2</u> , 25, 44, <u>131</u>	\endinput 10	N	\niib@bib . . . 137, <u>273</u>
C	\endnote 23	\niib@bibnote 122, 123, <u>223</u>	
\citenote <u>3</u> , <u>151</u>	\endnotemark 24	\niib@cite 225, 227, <u>228</u>	
D	\endnotetext 25	\niib@cnfalse 185, 232	
\delaycite <u>5</u> ,	F		
	\flushcitestack <u>4</u> , <u>140</u> , 281		

\niib@cntrue 230	211, 237, 238, 240	\niibpresorthead
\niib@delaycite 73, <u>186</u>	\niib@stream <u>252</u> <u>4</u> , <u>112</u> , 169
\niib@fnfalse 13	\niib@tailfalse 31, 62	\niibpresortnone
\niib@fnmark ... <u>36</u> , 40	\niib@tailtrue ... 33 <u>4</u> , <u>112</u> , 176
\niib@fntext ... <u>36</u> , 41	\niib@temp 207, 209, 211	\niibpresorttail
\niib@fntrue 15	\niib@text 132, 224, <u>251</u> <u>4</u> , <u>112</u> , 173
\niib@headfalse 32, 62	\niib@thanks <u>36</u> , 49, 53	\niibrecord <u>4</u> , <u>110</u> , 262
\niib@headtrue ... 34	\niibcite	
\niib@key <u>168</u> , 264	<u>4</u> , <u>118</u> , 156, 158,	P
\niib@keyname <u>165</u> , 264	163, 231, 244, 248	\printbibnotes ..
\niib@mark 128, <u>129</u> , <u>227</u>	\niibfield . <u>4</u> , <u>110</u> , 265 <u>3</u> , 27, <u>134</u>
\niib@msg <u>180</u> , 257	\niibkeyhead <u>4</u> , <u>115</u> , 170	
\niib@out . <u>252</u> , 261, 283	\niibkeynone <u>4</u> , <u>115</u> , 177	T
\niib@presort <u>168</u> , 263	\niibkeytail <u>4</u> , <u>115</u> , 174	\thanks <u>40</u>
\niib@stack .. 141–	\niibprefix . <u>4</u> , <u>109</u> ,	\thebibnote
143, <u>184</u> , 208, 209,	138, 255, 271, 276	<u>4</u> , <u>107</u> , 123, 129, 131

10 Notes

- [1] Note for the first example.
- [2] Note for the second example.
- [3] Note for the third example.
- [4] Some \verb-like output.