

# notes2bib — Integrating notes into the bibliography \*

Joseph Wright <sup>†</sup>

Released 2007/10/31

## 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

<b>1 Background</b>	<b>1</b>	<b>6 The mechanism</b>	<b>5</b>
<b>2 Basic use</b>	<b>2</b>	6.1 The <code>endnotes</code> approach	5
2.1 Package options . . . . .	2	6.2 The REVTeX approach .	6
2.2 Output of notes . . . . .	3	6.3 The <code>notes2bib</code> approach	6
2.3 Cross-referencing notes	3	<b>7 The package code</b>	<b>6</b>
2.4 Interaction with other packages . . . . .	3	7.1 Package setup code . . .	6
<b>3 Customisation</b>	<b>4</b>	7.2 User configuration macros	9
<b>4 Special effects</b>	<b>4</b>	7.3 User macros . . . . .	10
<b>5 Known issues</b>	<b>5</b>	7.4 Internal setup macros .	12
		7.5 Internal macros . . . . .	12
		<b>8 Change History</b>	<b>17</b>
		<b>9 Index</b>	<b>18</b>

## 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 v1.2a, last revised 2007/10/31.

<sup>†</sup>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 [?].      A very simple example of a bibliography note  
`\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 [?]. The      An example of a named note  
text can then continue and reference      `\bibnote[labelled-note]{Note for the second example}.`  
the note again later [?].      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  
`\bibnotetext` macros `\bibnotemark` and `\bibnotetext`. The text provided for each not is  
not “fragile,” and so it should not be necessary to use `\bibnotemark` directly.  
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      It is hard to write a good example for this `\bibnotemark!`  
for this [?]. The text continues      The text continues here `\bibnotetext{Note for the`  
here .      `third example}.`

### 2.1 Package options

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

<sup>1</sup>The `\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.<sup>2</sup> 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.<sup>3</sup> 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 `[?]` and using `[?]`.

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`.<sup>4</sup>

<sup>2</sup>The `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.

<sup>3</sup>LaTeX will complain if the user puts `\bibliography{}`.

<sup>4</sup>It 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.<sup>5</sup>

`notes2bib` attempts to maintain the correct ordering of notes with sorted  
bibliography styles. This is done by setting up suitable information in the  
temporary 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

<sup>5</sup>The `rs` 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`.<sup>6</sup> 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  $\langle macro-name \rangle$ , 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.<sup>7</sup> This is the same as for normal footnotes, and so the usual work-arounds are applicable.

The next note contains some awkward text [?].

The next note contains some awkward text

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

## 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.

<sup>6</sup>Notice that `\nocite` is not modified

<sup>7</sup>Actually, 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!

## 6.1 The endnotes approach

The `endnotes` package<sup>8</sup> 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 unexpanded to the file, the only requirement being that the braces match within the argument.<sup>9</sup>

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. The delay is brought about by altering the definition of `\if@filesw`, so that it is dependent on the context.

---

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

<sup>9</sup>Writing to the file uses the e-TeX `\unexpanded` primitive.

## 7 The package code

### 7.1 Package setup code

The package starts with the usual identification code.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{notes2bib}
3 [2007/10/31 v1.2a 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 etoolbox package is loaded to extend various LaTeX commands.

```
12 \RequirePackage{etoolbox}
```

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{%
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\relax

\niib@fnmark If the footnote option is active, there is some work to do because of the
\niib@fntext \thanks macro. This is all done using the etoolbox package. Firstly, it is nec-
\niib@thanks essary to back up the original definitions for \footnotetext, \footnotetext
and \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!\MessageBreak
62       bibnotes will be interspersed with other citations}
63     \niib@headfalse\niib@tailfalse
64   \else
65     \PackageInfo{notes2bib}%
66     {bibnotes will head other citations}
67   \fi
68 \else
69   \ifniib@tail
70     \PackageInfo{notes2bib}%

```



```

71      {bibnotes will tail other citations}
72  \else
73      \PackageInfo{notes2bib}%
74      {bibnotes will be interspersed with other citations}
75  \fi
76 \fi

```

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

```

77 \AtBeginDocument{%
78   \ifniib@head

```

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

```

79   \let\delaycite\niib@delaycite
80   \delaycite{cite}%

```

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

```

81   \@ifpackageloaded{natbib}{%
82     \delaycite{citet}%
83     \delaycite{citep}%
84     \delaycite{citenum}%
85     \delaycite{citealt}%
86     \delaycite{citealp}%
87     \delaycite{citetext}%
88     \delaycite{citeauthor}%
89     \delaycite{citeyear}%
90     \delaycite{citeyearpar}%
91     \delaycite{Citet}%
92     \delaycite{Citep}%
93     \delaycite{Citealt}%
94     \delaycite{Citealp}%
95     \delaycite{Citeauthor}%
96   }{}%

```

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

```

97   \@ifpackageloaded{biblatex}{%
98     \delaycite{Cite}%
99     \delaycite{parencite}%
100    \delaycite{Parencite}%
101    \delaycite{footcite}%
102    \delaycite{Footcite}%
103    \delaycite{textcite}%
104    \delaycite{Textcite}%

```

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

```

105    \@ifundefined{autocite}{}%
106    {\delaycite{autocite}}%
107    \@ifundefined{supercite}{}%
108    {\delaycite{supercite}}%
109  }{}%
110 \fi%
111 }

```

## 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.  112 <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.  113 <code>\newcounter{bibnote}</code> 114 <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> ).  115 <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.  116 <code>\newcommand*\niibrecord{Misc}</code> 117 <code>\newcommand*\niibfield{note}</code>
<code>\niibpresorthead</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.  118 <code>\newcommand*\niibpresorthead{ml}</code> 119 <code>\newcommand*\niibpresortnone{mm}</code> 120 <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.  121 <code>\newcommand*\niibkeyhead{aaa}</code> 122 <code>\newcommand*\niibkeynone{}</code> 123 <code>\newcommand*\niibkeytail{zzz}</code>
<code>\niibcite</code>	Bibliography notes are marked by citations, like any other bibliography entry. The citation command use is controlled here.  124 <code>\newcommand*\niibcite{\cite}</code>

### 7.3 User macros

<code>\bibnote</code>	Each new <code>bibnote</code> increments the note counter, then checks for an optional label, before handing off to the internal macro <code>\niib@bibnote</code> .  <pre> 125 \newrobustcmd*{\bibnote}{% 126   \stepcounter{bibnote}% 127   \@ifnextchar[% 128     {\niib@bibnote}% 129     {\niib@bibnote[\thebibnote]}}% 130 }</pre>
<code>\bibnotemark</code>	The <code>\bibnotemark</code> macro works in the same way as <code>\bibnote</code> , but calls <code>\niib@mark</code> rather than <code>\niib@bibnote</code> .  <pre> 131 \newrobustcmd*{\bibnotemark}{% 132   \stepcounter{bibnote}% 133   \@ifnextchar[% 134     {\niib@mark}% 135     {\niib@mark[\thebibnote]}}% 136 }</pre>
<code>\bibnotetext</code>	The text companion to the mark macro above, with no increment of the counter.  <pre> 137 \newrobustcmd*{\bibnotetext}[1][\thebibnote]{% 138   \niib@text{#1}% 139 }</pre>
<code>\printbibnotes</code>	To allow for the possibility of there being no other notes, a command to print only notes is given. In the <code>biblatex</code> case, the best that can be done is to issue <code>\printbibliography</code> .  <pre> 140 \@ifpackageloaded{biblatex}{% 141   \let\printbibnotes\printbibliography% 142 }{% 143   \newrobustcmd*{\printbibnotes}{\niib@bib% 144     {\niibprefix\jobname}}% 145 }</pre>
<code>\flushcitestack</code>	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 <code>.aux</code> file. This is done here, and the stack is cleared so collection can begin again.  <pre> 146 \newrobustcmd*{\flushcitestack}{% 147   \ifx\niib@stack\relax\else% 148     \expandafter\nocite\expandafter{\niib@stack}% 149     \global\let\niib@stack\relax% 150   \fi% 151 }</pre>
<code>\delaycite</code>	  <pre> 152 \newrobustcmd*{\delaycite}[1]{% 153   \PackageWarning{notes2bib}{% 154     \@nameuse{delaycite} can only be used in the document 155     body\MessageBreak Not modifying citation macro \@nameuse{#1}}% 156 }</pre>

`\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.

```
157 \ifniib@tail
158   \newrobustcmd*\citenote{%
159     \niib@cnfalse%
160     \niibcite%
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.

```
\niib@key
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
```

## 7.5 Internal macros

To alter citation ordering, tricks have to be played with writing to the .aux file. First some switches are needed to track what is going on. \ifniib@cn tracks is a note is being cited, \ifniib@citing is used to check for any citation, \ifniib@filesw is used to substitute \if@filesw, and \ifniib@orig@filesw records the original value of the file switch.

```
185 \newif\ifniib@cn\niib@cnfalse%
186 \newif\ifniib@citing\niib@citingfalse%
187 \newif\ifniib@filesw\niib@fileswfalse%
188 \newif\ifniib@orig@filesw
```

The value of the file switch is backed up.

```
189 \if@filesw
190   \niib@orig@fileswtrue%
191 \else
192   \niib@orig@fileswfalse%
193 \fi
```

\niib@head@filesw When in head mode, \if@filesw is modified using this macro.

```
194 \def\niib@head@filesw{%
```

First, the switching commands are altered.

```
195   \let\@fileswtrue\niib@orig@fileswtrue%
196   \let\@fileswfalse\niib@orig@fileswfalse%
```

The switch is now redefined. First, the current value of the original switch is transferred to the switch actually in use.

```
197   \def\if@filesw{%
198     \ifniib@orig@filesw
199     \niib@fileswtrue%
200   \else
201     \niib@fileswfalse%
202   \fi
```

If a citation is being made, and it is not a note, then writing to the .aux file is turned off.

```
203     \ifniib@citing
204       \ifniib@cn\else
205       \niib@fileswfalse%
206     \fi
207   \fi
```

The switches are reset, then the new switch is used to give the desired behaviour.

```
208   \global\niib@cnfalse%
209   \global\niib@citingfalse%
210   \ifniib@filesw%
211 }
212 }
```

If head mode is on, then this is all activated.

```
213 \ifniib@head
214   \niib@head@filesw%
215 \fi
```

`\niib@tail@files` When in tail mode, a slightly different set of modifications is needed.

```
216 \def\niib@tail@filesw{%
217   \let\@fileswtrue\niib@orig@fileswtrue%
218   \let\@fileswfalse\niib@orig@fileswfalse%
219   \def\if@filesw{%
220     \ifniib@orig@filesw
221       \niib@fileswtrue%
222     \else
223       \niib@fileswfalse%
224     \fi
```

If a note is being cited, writing to the `.aux` file is disabled.

```
225   \ifniib@cn
226     \niib@fileswfalse%
227   \fi
```

The switches are reset, then the new switch is used to give the desired behaviour.

```
228   \global\niib@cnfalse%
229   \ifniib@filesw%
230 }
231 }
```

If head mode is on, then this is all activated.

```
232 \ifniib@tail
233   \niib@tail@filesw%
234 \fi
```

`\niib@delaycite` The citation command has to be modified when in head mode. This is done here.

```
235 \newrobustcmd*\niib@delaycite[1]{%
236   \@ifundefined{#1}{%
237     \PackageWarning{notes2bib}
238       {Citation macro \@nameuse{#1} undefined\MessageBreak
239       \protect\delaycite\space cannot operate on this macro}
240   }{%
241     \@ifundefined{niib@orig@#1}{%
```

If the citation is not already modified, the original definition is backed up. It is then modified to ensure the correct behaviour of the file switch, and to stack up citations if needed.

```
242     \expandafter\expandafter\expandafter\let\expandafter%
243     \csname niib@orig@#1\expandafter\endcsname\csname #1\endcsname%
```

To ensure that the original citation command can see the next argument, a bit of complexity is needed. Two different definitions are used, depending on whether an optional argument is detected.

```
244     \expandafter\renewcommand\csname #1\endcsname{%
245       \niib@citingtrue%
246       \@ifnextchar[%]
247         {\@nameuse{niib@optarg@#1}}
248         {\@nameuse{niib@noarg@#1}}
249     }
```

There could be two optional arguments when using `biblatex`. So a second check is used.

```

250 \expandafter\def\csname niib@optarg@#1\endcsname[##1]{%
251 \ifnextchar[%
252 {\@nameuse{niib@twopt@#1}[##1]}
253 {\@nameuse{niib@oneopt@#1}[##1]}
254 }
255 \expandafter\def\csname niib@oneopt@#1\endcsname[##1]##2{%
256 \niib@stackup{##2}%
257 \@nameuse{niib@orig@#1}[##1]{##2}%
258 }
259 \expandafter\def\csname niib@twopt@#1\endcsname[##1][##2]##3{%
260 \niib@stackup{##3}%
261 \@nameuse{niib@orig@#1}[##1][##2]{##3}%
262 }

```

For no optional arguments, life is a bit easier!

```

263 \expandafter\def\csname niib@noarg@#1\endcsname##1{%
264 \niib@stackup{##1}%
265 \@nameuse{niib@orig@#1}{##1}%
266 }
267 }{%
268 \PackageInfo{notes2bib}%
269 {Already delayed \@nameuse{#1}}%
270 }
271 }
272 }

```

`\niib@stackup` The various optional argument tricks above all use the same core code, which adds the mandatory argument of the citation to the stack.

```

273 \def\niib@stackup#1{%
274 \ifniib@cn\else
275 \edef\niib@temp{#1}%
276 \ifx\niib@stack\relax%
277 \xdef\niib@stack{\niib@temp}%
278 \else
279 \xdef\niib@stack{\niib@stack,\niib@temp}%
280 \fi
281 \fi
282 }

```

`\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`).

```

283 \long\def\niib@bibnote[#1]##2{%
284 \niib@text{#1}{##2}%
285 \niib@cite{#1}%
286 }

```

`\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.

```

287 \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.

```

288 \ifniib@head%
289   \def\niib@cite#1{%
290     \niib@cntrue%
291     \niibcite{#1}%
292   }%

```

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.

```

293 \else%
294   \ifniib@tail%
295     \def\niib@cite#1{%
296       \ifx\niib@stack\relax%
297         \xdef\niib@stack{#1}%
298       \else%
299         \xdef\niib@stack{\niib@stack,#1}%
300       \fi%
301       \niib@cntrue%
302       \niibcite{#1}%
303     }%

```

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!

```

304   \else%
305     \def\niib@cite#1{\niibcite{#1}}%
306   \fi%
307 \fi%

```

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

```

308 \long\def\niib@text#1#2{%

```

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

\niib@stream

```

309   \@ifundefined{niib@out}{%
310     \if@files%
311       \newwrite\niib@out%
312       \gdef\niib@stream{\niibprefix\jobname.bib}%
313       \immediate\openout\niib@out\niib@stream\relax%

```

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

```

314     \immediate\write\niib@out{\niib@msg}%
315   \fi}%
316   {}%

```

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.

```

317   \if@files%
318     \immediate\write\niib@out{%
319       @\niibrecord\string{#1,^^J%
320       presort = \string{\niib@presort\string},^^J%
321       \niib@keyname\space= \string{\niib@key#1\string},^^J%

```



```

322     \niibfield\space= \string{\unexpanded{#2}\string}^^J%
323     \string}^^J}%
324 \fi%
325 }

```

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.

```

326 \@ifpackageloaded{biblatex}{%
327   \expandafter\bib@gadd\expandafter\blx@bibfiles\expandafter%
328   {,\niibprefix\jobname}%
329 }{%

```

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

```

\ Bibliography
330 \let\niib@bib\bibliography%
331 \renewcommand{\bibliography}[1]{%
332   \ifnum\the\value{bibnote} > \z@%
333     \niib@bib{\niibprefix\jobname,#1}%
334   \else%
335     \niib@bib{#1}%
336   \fi}%

```

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

```

337 \AtEndDocument{%
338   \flushcitestack%
339   \@ifundefined{niib@out}{}%
340   {\immediate\closeout\niib@out}%
341 }

```

## 8 Change History

v1.0		<code>\bibnote</code> : Macro made robust ..	10
General: Initial public release	...	<code>\bibnotemark</code> : Macro made robust .....	11
v1.0a		<code>\bibnotetext</code> : Macro made robust .....	11
<code>\citenote</code> : New macro .....	11	Use <code>\newrobustcmd</code> rather than <code>\@ifnextchar</code> for detecting optional argument .....	11
<code>\niib@delaycite</code> : Fixed a flaw with checking for existing redefinition .....	14	<code>\citenote</code> : Macro made robust ..	11
v1.1		<code>\delaycite</code> : Initial definition prevents use in preamble .....	11
General: <code>\Percent</code> macro removed .....	1	<code>\flushcitestack</code> : Macro made robust .....	11
Documentation improvements ..	1	<code>\niib@cite</code> : Definition made option-specific .....	15
Improvements to documentation and <code>dx</code> file .....	1	Name change from <code>\niib@citenote</code> .....	15
License changed from GPL to LPPL .....	1	<code>\niib@delaycite</code> : Added a san-	
Removed use of <code>xspace</code> .....	6		
Require e-TeX extensions .....	7		
Several code sections re-factored	1		

ity check on argument . . . . .	14	Name change of support package	
Improved logging of function . .	14	from elatex to etoolbox . . . .	7
Macro made robust . . . . .	14	\citenote: tail version altered	
Made internal to prevent execu-		to work with new delaying	
tion in preamble . . . . .	14	method . . . . .	11
\niib@text: Combined		\niib@cite: Moved resetting of	
\niib@text and \niib@@text		niib@cn flag . . . . .	15
. . . . .	16	\niib@delaycite: Significantly	
Improved awareness of files		altered to use new citation de-	
switch . . . . .	16	laying method . . . . .	14
\printbibnotes: Macro made ro-		\niib@head@files: New	
bust . . . . .	11	macro . . . . .	13
vi.2		\niib@stackup: New macro . . .	15
General: Altered implementation of		\niib@tail@files: New	
head and tail options to allow		macro . . . . .	13
moving of superscript citations .	1	vi.2a	
Fixed bug with biblatex sup-		General: Improved messages on	
port . . . . .	16	loading package . . . . .	8

## 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>Symbols</b>	\ifniib@files . .	\niib@fnfalse . . . .	13
\@fileswtrue . 195, 217	. . . . . 187, 210, 229	\niib@fnmark . .	<u>36</u> , 40
<b>B</b>	\ifniib@fn . . . .	\niib@fntext . .	<u>36</u> , 41
\bibliography . . .	<u>330</u>	\niib@fntrue . . . .	15
\bibnote . 2, 17, 23, <u>125</u>	\ifniib@head . .	\niib@head@files	
\bibnotemark . . . .	57, 78, 168, 213, 288	. . . . .	<u>194</u> , 214
. . . . . 2, 24, 43, <u>131</u>	\ifniib@orig@files	\niib@headfalse 32, 63	
\bibnotename 4, <u>112</u> , 114	. . . . . 188, 198, 220	\niib@headtrue . .	34
\bibnotetext . . . .	\ifniib@tail 31, 58,	\niib@key . . . .	<u>168</u> , 321
. . . . . 2, 25, 44, <u>137</u>	69, 157, 172, 232, 294	\niib@keyname <u>165</u> , 321	
<b>C</b>	<b>N</b>	\niib@mark 134, 135, <u>287</u>	
\citenote . . . . . 3, <u>157</u>	\niib@bib . . . .	\niib@msg . . . .	<u>180</u> , 314
<b>D</b>	. . . . . 128, 129, <u>283</u>	\niib@orig@filesfalse	
\delaycite 5, 79, 80,	\niib@cite 285, 287, <u>288</u>	. . . . . 192, 196, 218	
82–95, 98–104,	\niib@citingfalse	\niib@orig@files>true	
106, 108, <u>152</u> , 239	. . . . . 186, 209	. . . . . 190, 195, 217	
<b>F</b>	\niib@citingtrue 245	\niib@out <u>309</u> , 318, 340	
\flushcitestack .	\niib@cnfalse . .	\niib@presort <u>168</u> , 320	
. . . . . 4, <u>146</u> , 338	. . . . . 159, 185, 208, 228	\niib@stack 147–149,	
<b>I</b>	\niib@cntrue . 290, 301	<u>184</u> , 276, 277,	
\ifniib@citing 186, 203	\niib@delaycite 79, <u>235</u>	279, 296, 297, 299	
\ifniib@cn . . . . .	\niib@filesfalse	\niib@stackup . . .	
. . . . . 185, 204, 225, 274	. . . . . 187,	. . . . . 256, 260, 264, <u>273</u>	
	201, 205, 223, 226	\niib@stream . . . .	<u>309</u>
	\niib@files>true	\niib@tail@files	
	. . . . . 199, 221	. . . . .	<u>216</u> , 233

\niib@tailfalse	31, 63	\niibkeynone	4, <u>121</u> , 177	\niibrecord	4, <u>116</u> , 319
\niib@tailtrue	... 33	\niibkeytail	4, <u>121</u> , 174		
\niib@temp	275, 277, 279	\niibprefix	4, <u>115</u> ,	<b>P</b>	
\niib@text	138, 284, <u>308</u>		144, 312, 328, 333	\printbibnotes	..
\niib@thanks	<u>36</u> , 49, 53	\niibpresorthead		.....	3, 27, <u>140</u>
\niibcite	4, <u>124</u> , 160,		..... 4, <u>118</u> , 169		
	163, 291, 302, 305	\niibpresortnone		<b>T</b>	
\niibfield	. 4, <u>116</u> , 322		..... 4, <u>118</u> , 176	\thanks	..... <u>40</u>
\niibkeyhead	4, <u>121</u> , 170	\niibpresorttail		\thebibnote	.....
			..... 4, <u>118</u> , 173		4, <u>113</u> , 129, 135, 137