

notes2bib — Integrating notes into the bibliography *

Joseph Wright [†]

Released 2007/08/30

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

1 Background	1	6 The mechanism	5
		6.1 The <code>endnotes</code> approach	5
		6.2 The REVTeX approach	6
		6.3 The <code>notes2bib</code> approach	6
2 Basic use	2	7 The package code	6
2.1 Package options	2	7.1 Package setup code . . .	6
2.2 Output of notes	3	7.2 User configuration macros	9
2.3 Cross-referencing notes .	3	7.3 User macros	10
2.4 Interaction with other packages	3	7.4 Internal setup macros . .	12
		7.5 Internal macros	13
3 Customisation	3	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 v1.0, last revised 2007/08/30.

[†]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

<code>\bibnote</code>	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 <code>\footnote</code> type: <code>\bibnote</code> . This can be used in exactly the same way as a footnote, taking one mandatory argument <code>{\text}</code> . The <code>\text</code> will be made available as to the bibliography as a note (henceforth referred to as a bibnote). By default, each bibnote is given an automatically-generated label (see Section 3 for details of modifying the automatic system). However, <code>\bibnote</code> accepts an optional argument <code>[\label]</code> , which can be used to over-ride this. This is particularly useful when a note will be referenced several times.
<code>\bibnotemark</code> <code>\bibnotetext</code>	In common with <code>\footnote</code> , the basic <code>\bibnote</code> macro has companion macros <code>\bibnotemark</code> and <code>\bibnotetext</code> . The text provided for each not is not “fragile,” and so it should not be necessary to use <code>\bibnotemark</code> directly. It is needed when replacing footnotes by bibnotes. Notice that there <i>are</i> places where bibnotes will be problematic, for example in section headings which also appear in the Table of Contents. In these contexts, use <code>\cite</code> to reference the note, or use an optional argument to the <code>\section</code> , <i>etc.</i>

2.1 Package options

The `notes2bib` package currently accepts a number of options. The `footnotes` option turns standard LaTeX footnotes into bibnotes [1]. Thus footnotes then 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.

Finally, the package takes the `killpercent` option. This affects the handling of percentage characters within notes. By default, the percentage character is treated as a comment marker (as normal). Thus the rest of the line is ignored.

With the `killpercent` option, the `%` is completely skipped over, *e.g.* writing `Some_text_%_A_comment` will place `Some_text_A_comment` in the notes database [3].

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 [4]. 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 [5]. 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

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. This should be done using the `\cite` command (or similar) as the labels are bibliographic citations.

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` [6].

The `notes2bib` package is compatible with the current release of `biblatex` (v0.6). As `biblatex` works very differently from other citation systems, `notes2bib` must be loaded *after* `biblatex`. 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`
`\thebibnote`

Each note created by `notes2bib` is automatically given a label, unless one 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.

<code>\niibcite</code>	The macro used to cite notes is set in <code>\niibcite</code> . By default this is <code>\cite</code> , but can be altered, for example <code>\renewcommand{\niibcite}{\citep}</code> if using <code>natbib</code> .
<code>\niibprefix</code>	The temporary BibTeX database used to store the notes has the default name <code>niib-\jobname.bib</code> . The prefix (“niib-”) can be altered by redefining <code>\niibprefix</code> .
<code>\niibrecord</code> <code>\niibfield</code>	When the notes database is written, each record is saved as type <code>Misc</code> , and the text is stored in a <code>note</code> field. Both of these can be altered by redefining <code>\niibrecord</code> and <code>\niibfield</code> . This could be useful with a bibliography style that has an explicit “Notes” record type, for example [7].
	<code>notes2bib</code> 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.
<code>\niibkeyhead</code> <code>\niibkeynone</code> <code>\niibkeytail</code>	For users of standard bibliography styles, this is achieved by generating a <code>key</code> field in the database, containing the label used for the note. When using the <code>head</code> or <code>tail</code> option, this is prefixed by <code>aaa</code> or <code>zzz</code> , respectively. The prefix can be altered by redefining <code>\niibkeyhead</code> or <code>\niibkeytail</code> , for example when numerical labels need to be taken into account. The package also provides a <code>\niibkeynone</code> macro, which defaults to an empty prefix.
<code>\niibpresorthhead</code> <code>\niibpresortnone</code> <code>\niibpresorttail</code>	When using <code>biblatex</code> , the <code>presort</code> field can be used to control sorting of the citations. Again, three macros <code>\niibpresorthhead</code> , <code>\niibpresortnone</code> and <code>\niibpresorttail</code> are provided, which are used in the correct context. The default values of <code>ml</code> , <code>mm</code> and <code>mn</code> , respectively, are designed to achieve the desired effect while allowing the user full control of sorting.

4 Special effects

<code>\Percent</code>	As explained in the description of the <code>killpercent</code> option, the default behaviour of the package is to treat the % sign as a comment marker, as normal. However, no “escape” is available while writing the note to file, and so <code>\%</code> will also start a comment. The package therefore provides the <code>\Percent</code> macro to produce the percentage sign in notes [8].
<code>\flushcitestack</code>	When using the <code>head</code> or <code>tail</code> options, citations are added to a stack as they are made. This stack is then flushed to the <code>.aux</code> file at the end of the document. If references are given by chapter (or other unit), this may not give the desired effect. The <code>\flushcitestack</code> 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.
<code>\delaycite</code>	When using the <code>head</code> option, normal citations have to be saved to the stack described above. This requires the modification of the citation commands. By default, <code>notes2bib</code> modifies the <code>\cite</code> command and most of the standard citation commands of <code>natbib</code> and <code>biblatex</code> [9]. However, if problems are encountered with citation order, the user can use the <code>\delaycite</code> macro to patch any arbitrary citation command. This macro takes a single argument <code>{\langle macro-name \rangle}</code> , which is the name of the citation command to modify <i>minus</i> the leading backslash. For example, the <code>\cite</code> command is patched using <code>\delaycite{\cite}</code> . The <code>\delaycite</code> 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

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 “`\mycite`” command):

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

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

Users of `cite` will find that `notes2bib` does not move citations for notes to after punctuation. The note will be cited where it is given in the source.

`biblatex` does not like notes that contain verbatim text. This is due to the methods `biblatex` employs to work. Currently, there does not seem to be a great deal `notes2bib` can do about this.

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` package [10] 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 [11]. Notice that by default the `%` character still acts as a comment marker. Percentage signs in the notes database would cause problems on reading the data back in after running BibTeX.

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   [2007/08/30 v1.0 Integrating notes into the bibliography]
4 \RequirePackage{xspace}
```

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.

```

5 \newif\ifniib@fn\niib@fnfalse
6 \DeclareOption{footnotes}{%
7   \niib@fntrue
8   \AtBeginDocument{%
9     \let\footnote\bibnote%
10 }}
11 \DeclareOption{endnotes}{%
12   \AtBeginDocument{%
13     \let\endnote\bibnote%
14     \let\endnotemark\bibnotemark%
15     \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.

```

16   \@ifpackageloaded{endnotes}%
17     {\let\theendnotes\printbibnotes}{}%
18 }

```

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.

```

19 \newif\ifniib@tail\niib@tailfalse
20 \newif\ifniib@head\niib@headfalse
21 \DeclareOption{tail}{\niib@tailtrue}
22 \DeclareOption{head}{\niib@headtrue}

```

`\niib@percent` For handling percentage signs in notes, an option is needed to alter handling.

```

23 \def\niib@percent{\catcode`\%=14\relax}
24 \DeclareOption{killpercent}{%
25   \def\niib@percent{\catcode`\%=9\relax}
26 }

```

For `.dtx` files, the `killpercent` option is necessary.

```

27 \@ifclassloaded{ltxdoc}{\ExecuteOptions{killpercent}}{}

```

Finally, the options are put into effect.

```

28 \ProcessOptions

```

If the `footnote` option is active, there is some work to do because of the `\thanks` macro.

```

29 \ifniib@fn

```

The `\thanks` macro can only be patched if e-TeX extensions are available. This is because `elatex` is used for the actual patching.

```

30   \@ifundefined{eTeXversion}{%
31     \PackageWarning{notes2bib}{%
32       e-TeX extensions unavailable\MessageBreak
33       \protect\footnotemark and \protect\footnotetext%
34       \MessageBreak unmodified}%
35   }{}

```

\niib@fnmark With e-TeX available, the `elateX` package can be used to do the patching needed.
 \niib@fntext Firstly, it is necessary to back up the original definitions for `\footnotetext` and
 \niib@thanks `\footnotetext`. The `\thanks` macro is also backed up.

```
36 \let\niib@fnmark\footnotemark
37 \let\niib@fntext\footnotetext
38 \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.

```
39 \RequirePackage{elateX}
40 \patchcmd{\thanks}{\footnotemark}{\niib@fnmark}{%
41 \patchcmd{\thanks}{\footnotetext}{\niib@fntext}{%
42 \AtBeginDocument{%
43 \let\footnotemark\bibnotemark
44 \let\footnotetext\bibnotetext
45 }%
46 }%
```

Things have gone wrong if the following lines are reached. So everything is put back to normal.

```
47 {\let\thanks\niib@thanks%
48 \PackageWarning{notes2bib}%
49 {Could not patch \protect\thanks}}%
50 }{%
51 \let\thanks\niib@thanks%
52 \PackageWarning{notes2bib}%
53 {Could not patch \protect\thanks}}%
54 }%
55 \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.

```
56 \ifniib@head
57 \ifniib@tail
58 \PackageWarning{notes2bib}
59 {You've given both ``head'' and ``tail'' as
60 options\MessageBreak
61 I'm ignoring both!}
62 \niib@headfalse\niib@tailfalse
63 \fi
64 \fi
```

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

```
65 \AtBeginDocument{%
66 \ifniib@head
```

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

```
67 \delaycite{cite}%
```

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

```
68 \@ifpackageloaded{natbib}{%
```



```

69      \delaycite{citet}%
70      \delaycite{citep}%
71      \delaycite{citenum}%
72      \delaycite{citealt}%
73      \delaycite{citealp}%
74      \delaycite{citetext}%
75      \delaycite{citeauthor}%
76      \delaycite{citeyear}%
77      \delaycite{citeyearpar}%
78      \delaycite{Citet}%
79      \delaycite{Citep}%
80      \delaycite{Citealt}%
81      \delaycite{Citealp}%
82      \delaycite{Citeauthor}%
83  }{%

```

For biblatex, some citation commands are always defined.

```

84      \@ifpackageloaded{biblatex}{%
85          \delaycite{Cite}%
86          \delaycite{parencite}%
87          \delaycite{Parencite}%
88          \delaycite{footcite}%
89          \delaycite{Footcite}%
90          \delaycite{textcite}%
91          \delaycite{Textcite}%

```

Now there are several citation commands that may be defined. As biblatex requires e-TeX, the `\ifdefined` primitive can be used.

```

92      \ifdefined\autocite%
93          \delaycite{autocite}%
94      \fi%
95      \ifdefined\supercite%
96          \delaycite{supercite}%
97      \fi%
98  }{%}%
99  }%
100 \fi%
101 }

```

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.

`\bibnotename` By default, all of the new citations take the prefix “Bibnote.” The default can be altered, and this is set up here.

```
102 \newcommand*\bibnotename{Bibnote}
```

`\thebibnote` A counter is needed for the notes created. In analogy to other counters in LaTeX, this is given a `\the...` name. The user should not really need to use this macro, but convention dictate that it has a user-space name.

```
103 \newcounter{bibnote}
```

```
104 \renewcommand*\thebibnote{\bibnotename\the\value{bibnote}}
```

<code>\niibprefix</code>	<p>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>).</p> <pre>105 \newcommand*\niibprefix{niib-}</pre>
<code>\niibrecord</code> <code>\niibfield</code>	<p>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.</p> <pre>106 \newcommand*\niibrecord{Misc} 107 \newcommand*\niibfield{note}</pre>
<code>\niibpresorthhead</code> <code>\niibpresortnone</code> <code>\niibpresorttail</code>	<p>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.</p> <pre>108 \newcommand*\niibpresorthhead{ml} 109 \newcommand*\niibpresortnone{mm} 110 \newcommand*\niibpresorttail{mn}</pre>
<code>\niibkeyhead</code> <code>\niibkeynone</code> <code>\niibkeytail</code>	<p>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.</p> <pre>111 \newcommand*\niibkeyhead{aaa} 112 \newcommand*\niibkeynone{} 113 \newcommand*\niibkeytail{zzz}</pre>
<code>\niibcite</code>	<p>Bibliography notes are marked by citations, like any other bibliography entry. The citation command use is controlled here.</p> <pre>114 \newcommand*\niibcite{\cite}</pre>

7.3 User macros

<code>\bibnote</code>	<p>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>.</p> <pre>115 \DeclareRobustCommand\bibnote{% 116 \stepcounter{bibnote}% 117 \@ifnextchar[% 118 {\niib@bibnote}% 119 {\niib@bibnote[\thebibnote]}% 120 }</pre>
<code>\bibnotemark</code>	<p>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>.</p> <pre>121 \DeclareRobustCommand\bibnotemark{% 122 \stepcounter{bibnote}% 123 \@ifnextchar[% 124 {\niib@mark}% 125 {\niib@mark[\thebibnote]}% 126 }</pre>

<code>\bibnotetext</code>	<p>The text companion to the mark macro above, with no increment of the counter.</p> <pre> 127 \DeclareRobustCommand\bibnotetext{% 128 \@ifnextchar[% 129 {\niib@text}% 130 {\niib@text[\thebibnote]}}% 131 }</pre>
<code>\printbibnotes</code>	<p>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 <code>\printbibliography</code>.</p> <pre> 132 \@ifpackageloaded{biblatex}{% 133 \let\printbibnotes\printbibliography% 134 }{% 135 \newcommand*{\printbibnotes}{\niib@bib% 136 {\niibprefix\jobname}}% 137 }%</pre>
<code>\flushcitestack</code>	<p>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.</p> <pre> 138 \newcommand*\flushcitestack{% 139 \ifx\niib@stack\relax\else% 140 \expandafter\nocite\expandafter{\niib@stack}% 141 \global\let\niib@stack\relax% 142 \fi% 143 }</pre>
<code>\delaycite</code>	<p>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 <code>\@filesw</code> to false while the citation takes place.</p> <pre> 144 \newcommand*\delaycite[1]{% 145 \expandafter\ifx\csname niib@orig@#1\endcsname\@undefined% 146 \expandafter\expandafter\expandafter\let\expandafter% 147 \csname niib@orig@#1\endcsname\expandafter\endcsname\csname #1\endcsname% 148 \expandafter\renewcommand\csname #1\endcsname[2][\relax]{% 149 \ifniib@cn% 150 \csname niib@orig@#1\endcsname{##2}% 151 \else% 152 \begingroup% 153 \@fileswfalse% 154 \ifx##1\relax% 155 \csname niib@orig@#1\endcsname{##2}% 156 \else% 157 \csname niib@orig@#1\endcsname[##1]{##2}% 158 \fi%</pre> <p>If a note is being cited, then there will be no optional argument, and <code>##1</code> will expand to <code>\relax</code>.</p> <p>If an optional argument has been given, then <code>##1</code> will expand to something other than <code>\relax</code>. So the original citation command is called with the optional argument.</p>

```

159         \endgroup%
160         \edef\niib@temp{##2}%
161         \ifx\niib@stack\relax%
162             \xdef\niib@stack{\niib@temp}%
163         \else%
164             \xdef\niib@stack{\niib@stack,\niib@temp}%
165         \fi%
166     \fi%
167 }%
168 \else%
169     \PackageInfo{notes2bib}%
170     {Citation type ``#1'' already delayed}%
171 \fi%
172 }

```

`\Percent` Finally, a way of producing a percentage sign in note is needed. This is because inside the note, no method for escaping a comment is available.

```
173 \newcommand*{\Percent}{\%\xspace}
```

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.”

```

174 \@ifpackageloaded{biblatex}
175   {\def\niib@keyname{keysort}}
176   {\def\niib@keyname{key}}

```

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

`\niib@key`

```

177 \ifniib@head
178   \def\niib@presort{\niibpresorthead}
179   \def\niib@key{\niibkeyhead}
180 \else
181   \ifniib@tail
182     \def\niib@presort{\niibpresorttail}
183     \def\niib@key{\niibkeytail}
184   \else
185     \def\niib@presort{\niibpresortnone}
186     \def\niib@key{\niibkeynone}
187   \fi
188 \fi

```

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

```

189 \edef\niib@msg{%
190   This is an auxiliary file used by the 'notes2bib' package.^J%
191   This file may safely be deleted. It will be recreated as
192   required.^J}

```

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

```
193 \let\niib@stack\relax
```

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

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

7.5 Internal macros

`\niib@text` Here the label and the text for the note are separated out before writing to the bibliography file.

```
195 \long\def\niib@text[#1]{%
196   \niib@@text{#1}%
197 }
```

`\niib@bibnote` Very similar processing to `\niib@text` is carried out here, but in this case a citation is also added to the file.

```
198 \long\def\niib@bibnote[#1]{%
199   \niib@citenote{#1}%
200   \niib@@text{#1}%
201 }
```

`\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.

```
202 \def\niib@mark[#1]{\niib@citenote{#1}}
```

`\niib@citenote` To allow the notes to “tail” the real citations, a special version of the `\cite` command is needed that can allow delaying writing the `\citation` command to the aux file.

```
203 \def\niib@citenote#1{%
204   \niib@cntrue%
```

`\niib@temp` `\niib@temp` is needed to hold the expanded citation.

```
205   \edef\niib@temp{#1}%
```

If “tail”-ing is needed, the `\cite...` command is told not to write to the .aux file using `\@filesw`.

```
206   \ifniib@tail%
207     \begingroup%
208       \@fileswfalse%
209       \niibcite{#1}%
210     \endgroup%
```

The citation is added to the stack for writing later. `\niib@stack` will always expand to `\relax` if it is empty.

```
211   \ifx\niib@stack\relax%
212     \xdef\niib@stack{\niib@temp}%
213   \else%
214     \xdef\niib@stack{\niib@stack,\niib@temp}%
215   \fi%
216   \else%
```

When tailing is not in operation, writing takes place as normal. The run of `\expandafters` is needed to close the `\if` cleanly.

```
217   \expandafter\niibcite\expandafter%
218   {\expandafter\niib@temp\expandafter}%
219   \fi%
```

```

220 \niib@cnfalse%
221 }%

```

The next set of macros provides the direct writing of the unexpanded note to file. The code is a modified from a usenet post by Ulrich Diez at `comp.text.tex`, but with everything stripped down to do just what is needed here. The package author takes responsibility for the code eating your socks!

```

\niib@verbtex The two macros here set up the necessary catcode changes and grouping. Notice
\niib@@verbtex that they do not have the note text as an argument. That is left to \niib@tempa
\niib@tempa
222 \newcommand\niib@verbtex[2][\ignorespaces]{%
223 \begingroup%
224 \let\do\@makeother\dospecials%
225 \catcode`\{=1\relax%
226 \catcode`\}=2\relax%
227 \niib@@verbtex{#1}{#2}%
228 }%
229 \newcommand\niib@@verbtex[2]{%
230 \begingroup%
231 \long\def\niib@tempa##1{%
232 \def\niib@tempa{##1}%
233 \@onelevel@sanitize\niib@tempa%
234 \expandafter\endgroup%
235 \expandafter\def%
236 \expandafter#2%
237 \expandafter{%
238 \niib@tempa}%
239 #1%
240 }%

```

New lines are set to be spaces.

```

241 \catcode`\^^M=10\relax%

```

Unlike the “real” unexpanded write, the catcode of % needs to be something that will be safe, so that comments do not cause problems. There is no escape character, so an alternative method for printing % is given elsewhere in the package. For `dtx` files and so on, this will not work. So an option is needed to ignore percentage signs entirely.

```

242 \niib@percent%
243 \expandafter\endgroup%
244 \niib@tempa%
245 }%

```

```

\niib@write This provides the access end of the unexpanded write. Once again, the note is not
an argument.

```

```

246 \newcommand*\niib@write{%
247 \@bsphack%
248 \begingroup%
249 \niib@verbtex[\niib@@write\ignorespaces]\niib@tempa%
250 }%

```

```

\niib@@write Writing actually takes place here. In order to get all of the code needed into
the .bib file, it is set out here. Thus this macro is the specialised part of the
mechanism.

```

```

251 \newcommand*\niib@@write[1]{%
252   \immediate\write\niib@out{%
253     @\niibrecord\string{\niib@tempb,^^J%
254     presort = \string{\niib@presort\string},^^J%
255     \niib@keyname\space= \string{\niib@key\niib@tempb\string},^^J%
256     \niibfield\space= \string{\niib@tempa\string}^^J%
257     \string}^^J}%
258   \endgroup%
259   \@esphack%
260 }%

```

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

```

261 \long\def\niib@@text#1{%

```

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

```

262   \ifx\niib@out\@undefined%
263     \newwrite\niib@out%
264     \gdef\niib@stream{\niibprefix\jobname.bib}%
265     \immediate\openout\niib@out\niib@stream\relax%

```

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

```

266     \immediate\write\niib@out{\niib@msg}%
267   \fi%

```

\niib@tempb As long files are being written, \niib@tempb is used to store the label, before calling the \niib@write macro to output the note text. \niib@write will pick up the next thing in the input, which should be the grouped note text.

```

268   \if@files%
269     \edef\niib@tempb{#1}%
270     \expandafter\niib@write%
271   \fi%
272 }

```

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.

```

273 \@ifpackageloaded{biblatex}{%
274   \expandafter\bib@gadd\expandafter\blx@bibfiles\expandafter%
275   {\niibprefix\jobname}%
276 }{%

```

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

```

277   \let\niib@bib\bibliography%
278   \renewcommand{\bibliography}[1]{%
279     \ifnum\the\value{bibnote} > \z@%
280       \niib@bib{\niibprefix\jobname,#1}%
281     \else%
282       \niib@bib{#1}%
283     \fi}%

```

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

```
284 \AtEndDocument{%
285   \flushcitestack%
286   \ifx\niib@out\@undefined\else%
287     \immediate\closeout\niib@out%
288   \fi%
289 }
```

8 Change History

v1.0

General: Initial public release 1

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.

Symbols	N	
\% 23, 25, 173	\niib@@text 196, 200, <u>261</u>	\niib@temp 160, 162, 164, <u>205</u>
\{ 225	\niib@@verbtex . <u>222</u>	\niib@tempa <u>222</u> , 249, 256
\} 226	\niib@@write . 249, <u>251</u>	\niib@tempb 253, 255, <u>268</u>
\^ 241	\niib@bib 135, <u>277</u>	\niib@text 129, 130, <u>195</u>
	\niib@bibnote 118, 119, <u>198</u>	\niib@thanks 36, 47, 51
B	\niib@citenote 199, 202, <u>203</u>	\niib@verbtex <u>222</u> , 249
\bibliography . . . <u>277</u>	\niib@cnfalse 194, 220	\niib@write . . 246, 270
\bibnote . . 1 , 9, 13, <u>115</u>	\niib@cntrue 204	\niibcite 3 , <u>114</u> , 209, 217
\bibnotemark 2 , 14, 43, <u>121</u>	\niib@fnfalse 5	\niibfield . 3 , <u>106</u> , 256
\bibnotename 3 , <u>102</u> , 104	\niib@fnmark . . . 36, 40	\niibkeyhead 4 , <u>111</u> , 179
\bibnotetext 2 , 15, 44, <u>127</u>	\niib@fntext . . . 36, 41	\niibkeynone 4 , <u>111</u> , 186
	\niib@fntrue 7	\niibkeytail 4 , <u>111</u> , 183
D	\niib@headfalse 20, 62	\niibprefix . 3 , <u>105</u> , 136, 264, 275, 280
\delaycite 4 , 67, 69– 82, 85–91, 93, 96, <u>144</u>	\niib@headtrue . . . 22	\niibpresorthhead 4 , <u>108</u> , 178
F	\niib@key 177, 255	\niibpresortnone 4 , <u>108</u> , 185
\flushcitestack 4 , <u>138</u> , 285	\niib@keyname <u>174</u> , 255	\niibpresorttail 4 , <u>108</u> , 182
\footnote 9	\niib@mark <u>124</u> , <u>125</u> , <u>202</u>	\niibrecord 3 , <u>106</u> , 253
	\niib@msg 189, 266	
I	\niib@out 252, <u>262</u> , 286, 287	P
\ifniib@cn . . . 149, 194	\niib@percent . <u>23</u> , 242	\Percent 4 , <u>173</u>
\ifniib@fn 5, 29	\niib@presort <u>177</u> , 254	\printbibnotes 2 , 17, <u>132</u>
\ifniib@head 20, 56, 66, 177	\niib@stack . . 139– 141, 161, 162, 164, <u>193</u> , 211, 212, 214	T
\ifniib@tail 19, 57, 181, 206	\niib@stream <u>262</u>	\thanks 38, <u>39</u>
	\niib@tailfalse 19, 62	\thebibnote 3 , <u>103</u> , 119, 125, 130
	\niib@tailtrue . . . 21	

10 Notes

- [1] The `\thanks` macro used in making titles uses the footnote mechanism. This is unaffected by loading `notes2bib` even with the `footnote` option.
- [2] 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.
- [3] When producing the documentation for the package, this was something of a headache until the problem was solved.
- [4] `biblatex` users will define a style when loading `biblatex`, of course.
- [5] LaTeX will complain if the user puts `\bibliography{}`.
- [6] It is usually the case that `hyperref` should be the very last package loaded in the preamble.
- [7] The `rsc` and `achemso` packages contain styles of this type.
- [8] There is no perfect way of getting unexpanded output from TeX. The `\unexpanded` e-TeX primitive, for example, adds extra spaces to control sequences. The package author thinks that having to type `\Percent` to get `%` is the least inconvenient solution.
- [9] Notice that `\nocite` is not modified.
- [10] <http://tug.ctan.org/macros/latex/contrib/misc/endnotes.sty>.
- [11] Many thanks to Ulrich Diez for the code to achieve the unexpanded writing.