

xnotes2bib — Integrating notes into the bibliography*

Joseph Wright[†]

Released 2008/10/31

Abstract

The `xnotes2bib` package is an *experimental* re-implementation of `notes2bib` using \LaTeX 3 commands internally (taken from the `expl3` bundle). The user interface is almost identical to `notes2bib`.

The `xnotes2bib` 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		5.1 Preliminaries	4
1 Introduction	2	5.2 User options	6
		5.3 User macros	8
2 Using the package	2	5.4 Macros for the .aux file . . .	10
2.1 Macros	2	5.5 Internal macros	11
2.2 Control values	2	5.6 Finalising	13
3 Testing the package	4		
4 Installation	4	Change History	17
5 Implementation	4	Index	18

*This file describes version v6, last revised 2008/10/31.

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

1 Introduction

The new code syntax introduced by $\LaTeX 3$ is very different from plain \TeX or $\LaTeX 2_{\epsilon}$. This means that converting packages to $\LaTeX 3$ requires working from the ground up if the new ideas are to be exploited fully. This package re-implements `notes2bib` using the experimental syntax for internal commands.

The internal logic of the package remains broadly similar to that in `notes2bib`. However, some opportunities to simplify the code or make the flow clearer have been taken. There is therefore no one-to-one correspondence between the internals of the two packages. In particular, this package does not guarantee backward-compatibility with `notes2bib` under all circumstances. $\LaTeX 3$ will require significant changes to documents, and so alterations to the logic of packages in order to make them more useful in the long term seem sensible.

2 Using the package

2.1 Macros

<code>\bibnote</code>	The user macros of <code>xnotes2bib</code> are intended to work in the same way as those in <code>notes2bib</code> . Please see <code>notes2bib.pdf</code> for details of:
<code>\bibnotemark</code>	
<code>\bibnotetext</code>	
<code>\flushnotestack</code>	
	<ul style="list-style-type: none">• <code>\bibnote</code>• <code>\bibnotemark</code>• <code>\bibnotetext</code>• <code>\flushnotestack</code>

The `\citenote` macro can be replaced by using `\bibnotemark` to cite the note for a second time.

2.2 Control values

<code>\niibsetup</code>	The main user changes between <code>notes2bib</code> and <code>xnotes2bib</code> are in the user options. These have been completely re-written here, to improve the logic of the interface and to remove unnecessary options. All package control is currently carried out using the <code>\niibsetup</code> macro, as the $\LaTeX 2_{\epsilon}$ kernel expands options and removes spaces. As in <code>notes2bib</code> , the <code>\niibsetup</code> macro accepts a key-value list as input.
-------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

There are four types of control key. Boolean keys take the values `true` and `false`. If the key name alone is given, this is the same as giving the `true` value. String keys save their value as a string for internal use: they can therefore take any input. Choice keys take

one of a range of pre-determined values, in a similar way to Boolean options. However, the list of possible values varies from key to key. Finally, macro keys require a macro as the value.

<code>convert endnotes</code> <code>convert footnotes</code>	To convert endnotes and footnotes into bibnotes, the options <code>convert endnotes</code> and <code>convert footnotes</code> are available. Both taken Boolean values, and are set to <code>false</code> when loading <code>xnotes2bib</code> . If endnotes are not defined elsewhere, the <code>convert endnotes</code> only defines them when set to <code>true</code> .
<code>placement</code>	The package is able to position notes in three ways in the bibliography. The standard method is to create the notes interspersed with “real” citations. The order in the source then determines the position in the bibliography. The package can also create notes such that they appear either before or after normal citations. The behaviour here is controlled by the <code>placement</code> option, which takes a choice from the list <code>after</code> , <code>before</code> and <code>mixed</code> . The <code>after</code> option places notes after normal citations, <code>before</code> places notes first and <code>mixed</code> mixes notes and citations based on the order in the source.
<code>key after</code> <code>key before</code> <code>key mixed</code>	Most sorted bibliography styles provide methods for controlling the sort order of entries. With standard <code>BIB_TE_X</code> styles, the <code>key</code> field is used for this purpose. The options <code>key after</code> , <code>key before</code> and <code>key mixed</code> are used to set this up. They contain an item to be added before the note label when creating the <code>key</code> field. The standard values are <code>aaa</code> for <code>key before</code> and <code>zzz</code> for <code>key after</code> : <code>key mixed</code> is empty. When using <code>biblatex</code> , <code>presort</code> field is available. The values used by <code>xnotes2bib</code> for this field are stored as <code>presort after</code> , <code>presort before</code> and <code>presort mixed</code> , with default values <code>mn</code> , <code>m1</code> and <code>mm</code> , respectively. The <code>presort</code> system is described in the <code>biblatex</code> manual.
<code>presort after</code> <code>presort before</code> <code>presort mixed</code>	
<code>write key field</code>	When using the package with <code>natbib</code> , it may be necessary to avoid writing the <code>key</code> field to the temporary database. This is seen with the style <code>unsrtnat</code> , for example. The Boolean option <code>write key field</code> is available to turn off writing the <code>key</code> field under these circumstances.
<code>cite</code>	As <code>xnotes2bib</code> uses <code>BIB_TE_X</code> to create notes, it uses a citation command to mark the notes in the body of the text. The command used is controlled by the macro <code>key cite</code> . The standard value is <code>\cite</code> .
<code>field name</code> <code>prefix</code> <code>record</code>	Four control values are available to affect low-level aspects of the creation of the notes database. The <code>prefix</code> option sets the string used by <code>xnotes2bib</code> to name the database, and is set to <code>niib-</code> initially. The <code>field</code> and <code>record</code> options contain the name of the field used for the note, and the <code>BIB_TE_X</code> record type used. The standard values are <code>note</code> and <code>Misc</code> , respectively. Finally, the <code>name</code> option sets the text used in addition to a number when automatically labelling notes. This is set to <code>Bibnote</code> : there is no space between this text and the number as this will confuse <code>BIB_TE_X</code> .
<code>debug</code>	The <code>debug</code> Boolean option is available for understanding what the package is doing. When set to <code>true</code> , the package writes additional information to the log to aid problem solving.

3 Testing the package

This release of `xnotes2bib` has several aims, all related to testing. Firstly, it provides a method to examine new ideas for `notes2bib`: feedback on the altered options is welcome. Second, it is a test-bed for using `expl3` to create packages. People familiar with the current $\text{\LaTeX}3$ ideas are encouraged to read the source and make suggestions. The new coding environment requires a more structured approach than $\text{\LaTeX}2_{\epsilon}$, and so it is possible ideas have been missed.

The author wonders if this package could be implemented using the `template` system and `xfootnotes`. Suggestions in this area are welcome!

In order to use `xnotes2bib`, you will need an up-to-date \TeX system and recent copies of `expl3` and `xparse`. The package is tested with the current SVN release of the $\text{\LaTeX}3$ packages, but these may change on a day-to-day basis. So testers should be prepared for problems such as undefined macros from the new kernel! The ϵ - \TeX extensions to \TeX are required by `xnotes2bib`.

Due to the experimental nature of the package, it currently has no version number. The package date should be used to check if versions are identical. That said, the version of `notes2bib` will be stepped when new copies of `xnotes2bib` are created.

While the package is experimental, it should work properly as a replacement for `notes2bib`, with the proviso that the options are very different. Any problems of function should be reported as normal.

4 Installation

The entire bundle is supplied with the TDS-ready `.zip` file `notes2bib.tds.zip`. Simply unzip this into your local `texmf` tree and run your hash program (`texhash` for $\text{\TeX}Live$ or `initexmf -u` for $\text{MiK}\text{\TeX}$)

To extract the package `xnotes2bib.sty` from `xnotes2bib.dtx`, run `(pdf)\LaTeX` on the later. This will also create the documentation and `README.txt` file. Three `(pdf)\LaTeX` runs with `\write18` enabled will also build the index and table of contents in the `.pdf`.

5 Implementation

5.1 Preliminaries

The package starts by loading the basics of `expl3`. To keep things clear, first `l3names` is loaded to do the basic identification, and then everything else is started up. This is based on Morten Høgholm's `xfrac` package lead-off.

```

1 <*package>
2 \ProvidesExplPackage
3  {\filename}{\filedate}{\fileversion}{\filedescription}
4 \RequirePackage {expl3,keys3,xparse}

```

As the `expl3` syntax is in use, there is no need to worry about white space in the code. So there are very few % characters in the following. As far as possible, `expl3` or `xparse` macros are used here; however, a lot of the user-space $\LaTeX 2_\epsilon$ macros should still be valid in $\LaTeX 3$. There are also a few @ macros where things are tidied closely to the $\LaTeX 2_\epsilon$ kernel at present (for example `\@filesw`).

```

\c_niib_err_tlp Error messages are created.
\l_niib_package_option_tlp
5 \err_file_new:Nn \c_niib_err_tlp {xnotes2bib.err}
6 \tlp_new:Nn \l_niib_err_option_tlp {option}
7 \err_interrupt_new:NNNnnn \c_niib_err_tlp \l_niib_err_option_tlp 0
8  {All~package~control~should~be~given~using~the~\token_to_str:N~
9   \niibsetup~\text_put_sp:~macro}
10  {\err_help_return_or_X:}
11  {}
12 \err_file_close:N \c_niib_err_tlp

```

As the $\LaTeX 2_\epsilon$ kernel expands package options, for the moment `xnotes2bib` does not accept any load-time options.

```

13 \DeclareOption* {
14  \err_interrupt:NNw \c_niib_err_tlp \l_niib_err_option_tlp
15 }
16 \ProcessOptions \scan_stop:

```

```

\g_niib_note_int A counter for the automatically-created notes is needed. This is a global value (life will
get very complicated if not).

```

```

17 \int_new:N \g_niib_note_int

```

```

\niib_thenote: The counter needs to be expressed as a usable name. A check is made so that if there are
more than ten notes then the number is padded.

```

```

18 \def_new:Npn \niib_thenote: {
19  \l_niib_name_tlp
20  \num_compare:nNnT {\g_niib_totalnotes_int} > {\c_nine} {
21    \num_compare:nNnF {\g_niib_note_int} > {\c_nine} {
22      0
23    }
24  }
25  \int_to_arabic:n \g_niib_note_int
26 }

```

5.2 User options

`\l_niib_after_cite_bool` The placement option needs two Boolean values.

```
\l_niib_before_cite_bool
27 \bool_new:N \l_niib_after_cite_bool
28 \bool_new:N \l_niib_before_cite_bool
```

`\niib_convert_endnotes:` Some set up for other options.

```
\niib_convert_footnotes:
29 \let_new:NN \niib_convert_endnotes: \scan_stop:
30 \let_new:NN \niib_convert_footnotes: \scan_stop:
```

`\niib_cite:n` The cite key is stored.

```
31 \keys_manage_quick:n {
32   niib/.cd:,
33   cite/.store-in~tlp:N           = \niib_cite:n,
```

`\niib_debug:n` The debug option is implemented as a choice.

```
34 debug/.choice:,
35 debug/.default:n           = true,
36 debug/true/.code:n         = {
37   \def:Npn \niib_debug:n ##1 {
38     \err_info:nn {##1} {(xnotes2bib) \text_put_four_sp:}
39   }
40 },
41 debug/false/.code:n        = {
42   \let:NN \niib_debug:n \use_none:n
43 },
```

The `convert_footnotes` and `convert_endnotes` options work as choices.

```
44 convert~endnotes/.choices:nn = {true,false} {
45   \let:Nc \niib_convert_endnotes:
46     {niib_convert_endnotes_ \l_keys_current_choice_tlp}
47 },
48 convert~endnotes/.default:n   = true,
49 convert~footnotes/.choices:nn = {true,false} {
50   \let:Nc \niib_convert_footnotes:
51     {niib_convert_footnotes_ \l_keys_current_choice_tlp}
52 },
53 convert~footnotes/.default:n   = true,
```

`\l_niib_field_tlp` Some storage macros.

```
\l_niib_key_after_tlp
\l_niib_key_before_tlp 54 /keys/current~module:n           = niib,
\l_niib_key_mixed_tlp
\l_niib_keyword_tlp
\l_niib_name_tlp
```

```

55 field/.store-in~t1p:c          = field,
56 key~after/.store-in~t1p:c      = key_after,
57 key~before/.store-in~t1p:c     = key_before,
58 key~mixed/.store-in~t1p:c      = key_mixed,
59 keyword/.store-in~t1p:c        = keyword,
60 name/.store-in~t1p:c           = name,

```

Another choice for placement.

```

61 placement/.choice:             = \l_niib_placement_t1p,
62 placement/.default:n           = mixed,
63 placement/after/.code:n        = {
64   \bool_set_true:N \l_niib_after_cite_bool
65   \bool_set_false:N \l_niib_before_cite_bool
66 },
67 placement/before/.code:n       = {
68   \bool_set_false:N \l_niib_after_cite_bool
69   \bool_set_true:N \l_niib_before_cite_bool
70 },
71 placement/mixed/.code:n        = {
72   \bool_set_false:N \l_niib_after_cite_bool
73   \bool_set_false:N \l_niib_before_cite_bool
74 },

```

```

\l_niib_prefix_t1p
\l_niib_presort_after_t1p
\l_niib_presort_before_t1p
\l_niib_presort_mixed_t1p
\l_niib_record_t1p
75 /keys/current~module:n        = niib,
76 prefix/.store-in~t1p:c        = prefix,
77 presort~after/.store-in~t1p:c = presort_after,
78 presort~before/.store-in~t1p:c = presort_before,
79 presort~mixed/.store-in~t1p:c = presort_mixed,
80 record/.store-in~t1p:c        = record,

```

\l_niib_write_key_bool One real Boolean key.

```

81 write~key~field/.boolean:c     = write_key
82 }

```

All of the default values are set.

```

83 \keys_manage_quick:n{
84   /niib/.cd:,
85   cite          = \cite,
86   debug         = false,
87   field         = note,
88   key~after     = zzz,
89   key~before    = aaa,
90   key~mixed     = {},
91   name          = Bibnote,

```

```

92 placement      = mixed,
93 prefix          = niib-,
94 presort~after   = mn,
95 presort~before  = ml,
96 presort~mixed   = mm,
97 record          = Misc,
98 write~key~field = true}

```

5.3 User macros

`\bibnote` [#1]: label
#2: text

The main user macro has to check if the automatic note number is needed. If it is, the optional argument is note used to `\bibnotetext` as this automatically assumes the current value of `\niib_thenote:`. The note text must be sorted out before the mark is made, as otherwise packages such as `cite` do not work correctly with punctuation. As the package is designed for $\text{\LaTeX}3$, it is assumed \epsilon-TeX is available and so `\exp_not:n` can be used on the mandatory argument. Hence everything is absorbed here without worrying about category codes.

```

99 \NewDocumentCommand {\bibnote} {o>{P}m} {
100   \IfNoValueTF {#1} {
101     \int_gincr:N \g_niib_note_int
102     \bibnotetext {#2}
103     \bibnotemark [\niib_thenote:]
104   }{
105     \bibnotetext [#1] {#2}
106     \bibnotemark [#1]
107   }
108 }

```

`\bibnotemark` [#1]: label

As with `\bibnote`, if no optional argument is given the global note tracking counter is incremented before carrying out the citation.

```

109 \NewDocumentCommand {\bibnotemark} {o} {
110   \IfNoValueTF {#1} {
111     \int_gincr:N \g_niib_note_int
112     \niib_notemark:n {\niib_thenote;}
113   }{
114     \niib_notemark:n {#1}
115   }
116 }

```

`\bibnotetext` [#1]: label

#2: text

Writing the note text to the database should only occur if files are being written. For the moment, this uses the $\LaTeX 2\epsilon$ flag (and so the tradition syntax).

```
117 \NewDocumentCommand {\bibnotetext} {0{\niib_thenote:}>{P}m} {
118   \if@filesw
```

`\g_niib_out_stream` A new output stream may need to be created, if this is the first note to write. The $\LaTeX 2\epsilon$ file switch is checked to see if this should occur.

```
119   \cs_if_free:NT \g_niib_out_stream {
120     \iow_new:N \g_niib_out_stream
121     \iow_open:Nn \g_niib_out_stream
122       {\l_niib_prefix_tlp \c_job_name_tlp.bib}
123     \iow_expanded:Nn \g_niib_out_stream
124       {This-is-an-auxiliary-file-used-by-the-'xnotes2bib'-package.^J
125         This-file-may-safely-be-deleted.^J It-will-be-recreated-as-
126         required.^J}
127   }
```

The writing macro works in expanded mode, as everything except the note text does need expansion. Notice that `\space` has to be used here after the macro names, as `~` does not work.

```
128   \niib_debug:n {
129     Writing~text:\MessageBreak #2\MessageBreak to-database-for-note-
130     '#1'
131   }
132   \iow_expanded:Nn \g_niib_out_stream {
133     @\l_niib_record_tlp{
134       #1,^J
135       \text_put_sp: \text_put_sp: \l_niib_field_tlp \text_put_sp:
136         =~{\exp_not:n{#2}},^J
137       \bool_if:NT \l_niib_write_key_bool {
138         \text_put_sp: \text_put_sp: \l_niib_keyname_tlp \text_put_sp:
139         =~{\niib_key:#1},^J
140       }
141       \text_put_sp: \text_put_sp: keywords~
142         =~{\l_niib_keyword_tlp},^J
143       \text_put_sp: \text_put_sp: presort~
144         =~{\niib_presort:},^J
145     }^J
146   }
147   \fi
148 }
```

`\flushnotestack` In order to delay citations to the end of the bibliography (and thus force others to the start), note citations are held in a clist until being added to the `.aux` file using `\flushnotestack`. At that point, the stack is cleared so that collection can begin again.

```

149 \NewDocumentCommand {\flushnotestack} {} {
150   \clist_if_empty:NF \g_niib_before_cite_clist {
151     \if@filesw
152       \iow_expanded:Nn \@auxout {
153         \token_to_str:N \niibbeforecite {
154           \g_niib_before_cite_clist
155         }
156       }
157     \fi
158   }
159   \clist_if_empty:NF \g_niib_after_cite_clist {
160     \nocite { \g_niib_after_cite_clist}
161     \if@filesw
162       \iow_expanded:Nn \@auxout {
163         \token_to_str:N \niibaftercite {\g_niib_after_cite_clist}
164       }
165     \fi
166   }
167   \clist_gclear:N \g_niib_after_cite_clist
168 }

```

`\niibsetup` The document-level command for settings.

```

169 \NewDocumentCommand {\niibsetup} {m} {\keys_manage:n {/niib/.cd:,#1}}

```

5.4 Macros for the .aux file

To allow tracking of the need to re-run \LaTeX , each run writes the current out-of-order citations to the .aux file. This requires two user-space macros, although it is not intended the user ever applies them!

`\g_niib_before_cite_old_clist` Two lists are created to track the out-of-order citations written in the previous run to the .aux file.
`\g_niib_after_cite_old_clist`

```

170 \clist_new:N \g_niib_before_cite_old_clist
171 \clist_new:N \g_niib_after_cite_old_clist

```

`\g_niib_document_tlp` A hook into the `\document` macro is needed by `\niibbeforecite`; this needs to exist before `\document` is executed, hence needing a `tlp` for this.

```

172 \tlp_new:N \g_niib_document_tlp
173 \tlp_gput_right:Nn \document {\g_niib_document_tlp}

```

`\niibbeforecite` #1: citations

`\niibaftercite` The macros themselves are quite simple. `\clist_gset_eq:Nc` is not used as the macros could appear more than once in the .aux file.

```

174 \NewDocumentCommand {\niibbeforecite} {m} {
175   \clist_gput_right:Nn \g_niib_before_cite_old_clist {#1}
176   \tlp_gput_right:Nn \g_niib_document_tlp {\nocite{#1}}
177 }
178 \NewDocumentCommand {\niibaftercite} {m} {
179   \clist_gput_right:Nn \g_niib_after_cite_old_clist {#1}
180 }

```

`\g_niib_totalnotes_int` The total number of notes in the last run is needed

```
181 \int_new:N \g_niib_totalnotes_int
```

`\niibttotalnotes` #1: number

The total number of notes created is stored for the next run.

```

182 \NewDocumentCommand {\niibttotalnotes} {m} {
183   \int_gset:Nn \g_niib_totalnotes_int {#1}
184 }

```

5.5 Internal macros

`\niib_notemark:n` The citation command itself is in an internal macro, as it is called from a few places. Both the before and after modes require some tracking of the citations.

```

185 \def_new:Npn \niib_notemark:n #1 {
186   \niib_debug:n {Creating~citation~for~note~‘#1’}
187   \bool_if:NT \l_niib_before_cite_bool {
188     \niib_debug:n {Note~‘#1’~before~real~citations}
189     \clist_gput_right:Nx \g_niib_before_cite_clist {#1}
190   }
191   \bool_if:NTF \l_niib_after_cite_bool {
192     \niib_debug:n {Note~‘#1’~after~real~citations}
193     \clist_gput_right:Nx \g_niib_after_cite_clist {#1}
194     \niib_after_cite:n {#1}
195   }{
196     \niib_cite:n {#1}
197   }
198 }

```

`\g_niib_before_cite_clist` Out-of-order citations in the current run are followed using two clists.

`\g_niib_after_cite_clist`

```

199 \clist_new:N \g_niib_before_cite_clist
200 \clist_new:N \g_niib_after_cite_clist

```

`\niib_key:` The key and presort information for the database needs to be created dynamically from
`\niib_presort:` the various keys.

```

201 \def_new:Npn \niib_key: {
202   \bool_if:NTF \l_niib_before_cite_bool {
203     \tlp_use:N \l_niib_key_before_tlp
204   }{
205     \bool_if:NTF \l_niib_after_cite_bool {
206       \tlp_use:N \l_niib_key_after_tlp
207     }{
208       \tlp_use:N \l_niib_key_mixed_tlp
209     }
210   }
211 }
212 \def_new:Npn \niib_presort: {
213   \bool_if:NTF \l_niib_before_cite_bool {
214     \tlp_use:N \l_niib_presort_before_tlp
215   }{
216     \bool_if:NTF \l_niib_after_cite_bool {
217       \tlp_use:N \l_niib_presort_after_tlp
218     }{
219       \tlp_use:N \l_niib_presort_mixed_tlp
220     }
221   }
222 }

```

`\niib_convert_endnotes_false:` Conversion of footnotes and endnotes takes place using four dedicated macros, which
`\niib_convert_endnotes_true:` can then be called by `csname`.

```

\niib_convert_footnotes_false:
\niib_convert_footnotes_true:
223 \def_new:Npn \niib_convert_endnotes_false: {
224   \niib_debug:n {Restoring~normal~endnotes}
225   \let:NN \endnote \niib_saved_endnote:
226   \let:NN \endnotemark \niib_saved_endnotemark:
227   \let:NN \endnotetext \niib_saved_endnotetext:
228 }
229 \def_new:Npn \niib_convert_endnotes_true: {
230   \niib_debug:n {Converting~endnotes~to~bibnotes}
231   \let:NN \endnote \bibnote
232   \let:NN \endnotemark \bibnotemark
233   \let:NN \endnotetext \bibnotetext
234 }
235 \def_new:Npn \niib_convert_footnotes_false: {
236   \niib_debug:n {Restoring~normal~footnotes}
237   \let:NN \footnote \niib_saved_footnote:
238   \let:NN \footnotemark \niib_saved_footnotemark:
239   \let:NN \footnotetext \niib_saved_footnotetext:
240 }
241 \def_new:Npn \niib_convert_footnotes_true: {
242   \niib_debug:n {Converting~footnotes~to~bibnotes}
243   \let:NN \footnote \bibnote
244   \let:NN \footnotemark \bibnotemark
245   \let:NN \footnotetext \bibnotetext
246 }

```

`\thanks` The `\thanks` macro has to be replaced by one that uses the original `\footnotemark` and `\footnotetext`.

```

247 \def:Npn \thanks #1 {
248   \niib_saved_footnotemark:
249   \protected@xdef \@thanks {
250     \@thanks
251     \protect \niib_saved_footnotetext: [\the\c@footnote] {#1}
252   }
253 }

```

5.6 Finalising

`\l_niib_keyname_tlp` The name used for the key field is stored a little later. The store is set up here.

```

254 \tlp_new:N \l_niib_keyname_tlp

```

`\niib_bibliography:n` Various macro definitions depend on the loading of other packages. When `biblatex` is in use, life is rather different to when the standard citation method is operational. The `\printbibnotes` macro is used to control sorting; `biblatex` uses a different macro name for this function than normal \LaTeX does. The `\printbibnotes` user macro always prints the bibliography, using only the notes database if possible. The `\niib_after_cite:n` macro has to keep a track of the appropriate bibnotes and also fiddle with the `\@filesw` switch to prevent writing at the wrong time.

```

255 \AtBeginDocument{
256   \err_info:nn {
257     (xnotes2bib) \text_put_sp: Beginning~\token_to_str:N
258     \AtBeginDocument \text_put_sp: tasks
259   } {}
260   \ifpackageloaded {biblatex} {
261     \tlp_set:Nn \l_niib_keyname_tlp {keysort}
262     \gappto \blx@bibfiles {,\niib_prefix\c_job_name_tlp}
263     \let_new:NN \printbibnotes \printbibliography
264     \def_new:Npn \niib_after_cite:n #1 {
265       \AtNextCite {\@fileswfalse}
266       \niib_cite:n {#1}
267     }
268   }{
269     \tlp_gset:Nn \l_niib_keyname_tlp {key}
270     \let_new:NN \niib_bibliography:n \bibliography
271     \def:Npn \bibliography #1 {
272       \int_compare:nNnTF {\g_niib_note_int} = {\c_zero} {
273         \niib_bibliography:n {#1}
274       }{
275         \niib_bibliography:n {
276           #1, \l_niib_prefix_tlp \c_job_name_tlp
277         }

```

```

278     }
279   }
280   \NewDocumentCommand {\printbibnotes} {} {
281     \niib_bibliography:n {\l_niib_prefix_tlp\c_job_name_tlp}
282   }

```

\ifniib@filesw There is a need to back-up the \@filesw switch. As this is a L^AT_EX₂_ε switch, the name and creation method used here are not L^AT_EX₃-based.

```

283   \newif \ifniib@filesw

```

\@restore@auxhandle If cite is loaded, there is further trouble.

```

\niib_auxhandle:
284   \@ifpackageloaded {cite}{
285     \def_new:Npn \niib_after_cite:n #1 {
286       \let:NN \ifniib@filesw \if@filesw
287       \@fileswfalse
288       \def:Npn \niib_auxhandle: {
289         \let:NN \if@filesw \ifniib@filesw
290         \let:NN \niib_auxhandle: \scan_stop:
291       }
292       \niib_cite:n {#1}
293     }
294     \tlp_gput_right:Nn \g_niib_document_tlp {
295       \cs_if_exist:NTF \@restore@auxhandle{
296         \tlp_put_right:Nn \@restore@auxhandle {\niib_auxhandle:}
297       }{
298         \def_new:Npn \@restore@auxhandle {\niib_auxhandle:}
299       }
300     }
301     \let_new:NN \niib_auxhandle: \scan_stop:

```

Life is slightly easier without cite.

```

302   }{
303     \def_new:Npn \niib_after_cite:n #1 {
304       \let:NN \ifniib@filesw \if@filesw
305       \@fileswfalse
306       \niib_cite:n {#1}
307       \let:NN \if@filesw \ifniib@filesw
308     }
309   }
310 }

```

\niib_saved_footnote: The second thing to worry about is footnotes and endnotes. The definitions are saved here so that any other packages can have made their changes. No argument specifiers are given as these are saving user space macros.

```

\niib_saved_endnote:
\niib_saved_endnotemark:
\niib_saved_endnotetext:
\niib_saved_endnote:
311 \let_new:NN \niib_saved_footnote: \footnote
\niib_saved_endnotemark:
\niib_saved_endnotetext:

```

```

312 \let_new:NN \niib_saved_footnotemark: \footnotemark
313 \let_new:NN \niib_saved_footnotetext: \footnotetext
314 \let_new:NN \niib_saved_endnote: \endnote
315 \let_new:NN \niib_saved_endnotemark: \endnotemark
316 \let_new:NN \niib_saved_endnotetext: \endnotetext

```

Any conversion can now be applied, before the convert footnotes and convert endnotes options are redefined to work immediately.

```

317 \niib_convert_endnotes:
318 \niib_convert_footnotes:
319 \keys_manage_quick:n {
320   /niib/.cd:,
321   convert~endnotes/true/.code:n = {\niib_convert_endnotes_true:},
322   convert~endnotes/false/.code:n = {\niib_convert_endnotes_false:},
323   convert~footnotes/true/.code:n = {\niib_convert_footnotes_true:},
324   convert~footnotes/false/.code:n = {\niib_convert_footnotes_false:},
325 }
326 \err_info:mn {
327   (xnotes2bib) \text_put_sp: Finished~\token_to_str:N
328   \AtBeginDocument \text_put_sp: tasks
329 } {}
330 }

```

`\niib_rerun_check:n` If either of the out-of-order options are in use, then a comparison of the citations used with those used in the last run is needed. There are a couple of easy checks to make: if only one list is empty, there must be changes. If both lists are occupied, a check of every item must be made.

```

331 \def_new:Npn \niib_rerun_check:n #1 {
332   \clist_if_empty:cTF {g_niib_old#1cite_clist}{
333     \clist_if_empty:cF {g_niib_#1cite_clist}{
334       \niib_rerun_log:
335     }
336   }{
337     \clist_if_empty:cTF {g_niib_#1cite_clist}{
338       \niib_rerun_log:

```

If both lists are occupied by data, the total contents of each one have to be compared with the other. This needs to be done both ways round.

```

339   }{
340     \bool_set_false:N \l_tmpa_bool
341     \def:Npn \niib_rerun_test:n ##1 {
342       \clist_if_in:cnTF {g_niib_old#1cite_clist} {##1} {
343         \bool_set_true:N \l_tmpa_bool
344         \clist_map_break:w
345       }{
346     }
347   }

```

```

348 \clist_map_function:cN {g_niib_#1cite_clist}
349 \niib_rerun_test:n
350 \def:Npn \niib_rerun_test:n ##1 {
351 \clist_if_in:cnTF {g_niib_#1cite_clist} {##1} {
352 \bool_set_true:N \l_tmpa_bool
353 \clist_map_break:w
354 }{
355 }
356 }
357 \clist_map_function:cN {g_niib_old#1cite_clist}
358 \niib_rerun_test:n
359 \bool_if:NF \l_tmpa_bool {
360 \niib_rerun_log:
361 }
362 }
363 }
364 }

```

`\niib_rerun_test:n` This is redefined in the above, but is made available here.

```

365 \let_new:NN \niib_rerun_test:n \use_none_i:n

```

`\niib_rerun_log:` The information message needs to be available in a few different places.

```

366 \def_new:Npn \niib_rerun_log: {
367 \err_info:nn {
368 (xnotes2bib) \text_put_sp: To-get-notes-in-the-correct-order,~
369 please \err_newline: rerun-LaTeX,~(re)run-BibTeX-on-the-file~
370 \c_job_name_tlp.aux \err_newline: and~rerun-LaTeX-again~afterwards
371 } {(xnotes2bib) \text_put_sp:}
372 \io_put_log:x {
373 REQ:3:latex:REQ \iow_newline:
374 REQ:2:bibtex:REQ \iow_newline:
375 REQ:1:latex:REQ \iow_newline:
376 }
377 \let \niib_rerun_log: \scan_stop:
378 }

```

At the end of the document, the package should tidy up.

```

379 \AtEndDocument{
380 \niib_rerun_check:n {before}
381 \niib_rerun_check:n {after}
382 \flushnotestack
383 \iow_expanded:Nn \@auxout {
384 \token_to_str:N \niibtotnotes {\int_to_arabic:n \g_niib_note_int}
385 }
386 \num_compare:nNnF {\g_niib_totnotes_int} = {\g_niib_note_int} {
387 \niib_rerun_log:

```



```

388 }
389 \cs_if_free:NF \g_niib_out_stream {
390   \iow_close:N \g_niib_out_stream
391 }
392 }
393 (/package)

```

Change History

v0.0		
\@restore@auxhandle: Moved patch code to after \AtBeginDocument ...	14	
General: Key management moved to l3keys	6	
Moved errors to new system	5	
Moved options to keys3	4	
Removed ε -TeX statement: this should be obvious to anyone using expl3	4	
Replaced \cs_use:c by \use:c	4	
Replaced \token_to_string:N with \token_to_str:N	4	
Replaced use of \def_new:NNn and \def:NNn with \def_new:Npn and \def:Npn, respectively	4	
\g_niib_after_cite_clist: Renamed from \g_niib_aftercite_clist ..	11	
\g_niib_after_cite_old_clist: Renamed from \g_niib_oldaftercite_clist	10	
\g_niib_before_cite_clist: Renamed from \g_niib_beforecite_clist	11	
\g_niib_before_cite_old_clist: Renamed from \g_niib_oldbeforecite_clist	10	
\g_niib_document_tlp: Changed \tlp_put_right:Nn to \tlp_gput_right:Nn	10	
\g_niib_out_stream: Replaced \space with \text_put_sp:	9	
		\l_niib_after_cite_bool: Renamed from \l_niib_after_bool
		6
		\l_niib_before_cite_bool: Renamed from \l_niib_before_bool
		6
		\l_niib_key_after_tlp: Renamed from \l_niib_key_after_tlp
		6
		\l_niib_key_before_tlp: Renamed from \l_niib_key_before_tlp ...
		6
		\l_niib_key_mixed_tlp: Renamed from \l_niib_key_mixed_tlp
		6
		\l_niib_keyname_tlp: Renamed from \niib_keyname:
		13
		\l_niib_name_tlp: Renamed from \l_niib_notename_tlp
		6
		\l_niib_presort_after_tlp: Renamed from \l_niib_presort_after_tlp
		7
		\l_niib_presort_before_tlp: Renamed from \l_niib_presort_before_tlp
		7
		\l_niib_presort_mixed_tlp: Renamed from \l_niib_presort_mixed_tlp
		7
		\l_niib_record_tlp: Renamed and recoded from \niib_KV_string:n ..
		7
		\niib_after_cite:n: Renamed from \niib_aftercite:n
		13
		\niib_notemark:n: Replaced \clist_put_right:Nx by \clist_gput_right:Nx
		11
		\niib_rerun_log:: Maximum of one message per run
		16
		\niibsetup: Converted to l3keys method
		10

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
<code>\@auxout</code>	152, 162, 383
<code>\@fileswfalse</code>	265, 287, 305
<code>\@ifpackageloaded</code>	260, 284
<code>\@restore@auxhandle</code>	284
<code>\@thanks</code>	249, 250
A	
<code>\AtBeginDocument</code>	255, 258, 328
<code>\AtEndDocument</code>	379
<code>\AtNextCite</code>	265
B	
<code>\bibliography</code>	270, 271
<code>\bibnote</code>	2, 99, 231, 243
<code>\bibnotemark</code>	2, 103, 106, 109, 232, 244
<code>\bibnotetext</code>	2, 102, 105, 117, 233, 245
<code>\blx@bibfiles</code>	262
<code>\bool_if:NF</code>	359
<code>\bool_if:NT</code>	137, 187
<code>\bool_if:NTF</code>	191, 202, 205, 213, 216
<code>\bool_new:N</code>	27, 28
<code>\bool_set_false:N</code>	65, 68, 72, 73, 340
<code>\bool_set_true:N</code>	64, 69, 343, 352
C	
<code>\c@footnote</code>	251
<code>\c_job_name_tlp</code>	122, 262, 276, 281, 370
<code>\c_niib_err_tlp</code>	5, 14
<code>\c_nine</code>	20, 21
<code>\c_zero</code>	272
<code>\cite</code>	85
<code>cite (option)</code>	3
<code>\clist_gclear:N</code>	167
<code>\clist_gput_right:Nn</code>	175, 179
<code>\clist_gput_right:Nx</code>	189, 193
<code>\clist_if_empty:cF</code>	333
<code>\clist_if_empty:cTF</code>	332, 337
<code>\clist_if_empty:NF</code>	150, 159
<code>\clist_if_in:cnTF</code>	342, 351
<code>\clist_map_break:w</code>	344, 353
<code>\clist_map_function:cN</code>	348, 357
<code>\clist_new:N</code>	170, 171, 199, 200
<code>convert endnotes (option)</code>	3
<code>convert footnotes (option)</code>	3
<code>\cs_if_exist:NTF</code>	295
<code>\cs_if_free:NF</code>	389
<code>\cs_if_free:NT</code>	119
D	
<code>debug (option)</code>	3
<code>\DeclareOption</code>	13
<code>\def:Npn</code>	37, 247, 271, 288, 341, 350
<code>\def_new:Npn</code>	18, 185, 201, 212, 223, 229, 235, 241, 264, 285, 298, 303, 331, 366
<code>\document</code>	173
E	
<code>\endnote</code>	225, 231, 314
<code>\endnotemark</code>	226, 232, 315
<code>\endnotetext</code>	227, 233, 316
<code>\err_file_close:N</code>	12
<code>\err_file_new:Nn</code>	5
<code>\err_help_return_or_X:</code>	10
<code>\err_info:mn</code>	38, 256, 326, 367
<code>\err_interrupt:NNw</code>	14
<code>\err_interrupt_new:NNNnnn</code>	7
<code>\err_newline:</code>	369, 370
<code>\exp_not:n</code>	136
F	
<code>\fi</code>	147, 157, 165
<code>field (option)</code>	3
<code>\filedate</code>	3
<code>\filedescription</code>	3
<code>\filename</code>	3
<code>\fileversion</code>	3
<code>\flushnotestack</code>	2, 149, 382
<code>\footnote</code>	237, 243, 311
<code>\footnotemark</code>	238, 244, 312
<code>\footnotetext</code>	239, 245, 313
G	
<code>\g_niib_after_cite_clist</code>	159, 160, 163, 167, 193, 199
<code>\g_niib_after_cite_old_clist</code>	170, 179

\g_niib_before_cite_clist	\l_niib_presort_after_tlp	75, 217
..... 150, 154, 189, 199	\l_niib_presort_before_tlp	75, 214
\g_niib_before_cite_old_clist ..	\l_niib_presort_mixed_tlp	75, 219
..... 170, 175	\l_niib_record_tlp	75, 133
\g_niib_document_tlp	\l_niib_write_key_bool	81, 137
..... 172, 176, 294	\l_tmpa_bool	340, 343, 352, 359
\g_niib_note_int	\let	377
..... 17, 21, 25, 101, 111, 272, 384, 386	\let:Nc	45, 50
\g_niib_out_stream	\let:NN	42,
..... 119, 389, 390	225, 226, 227, 231, 232, 233, 237, 238,	
\g_niib_totalnotes_int ..	239, 243, 244, 245, 286, 289, 290, 304, 307	
..... 20, 181, 183, 386	\let_new:NN	29, 30, 263,
\gappto	270, 301, 311, 312, 313, 314, 315, 316, 365	
262		
I		
\if@filesw .	118, 151, 161, 286, 289, 304, 307	
\ifniib@filesw	283, 286, 289, 304, 307	
\IfNoValueTF	100, 110	
\int_compare:nNnTF	272	
\int_gincr:N	101, 111	
\int_gset:Nn	183	
\int_new:N	17, 181	
\int_to_arabic:n	25, 384	
\io_put_log:x	372	
\iow_close:N	390	
\iow_expanded:Nn ..	123, 132, 152, 162, 383	
\iow_new:N	120	
\iow_newline:	373, 374, 375	
\iow_open:Nn	121	
K		
key after (option)	3	
key before (option)	3	
key mixed (option)	3	
\keys_manage:n	169	
\keys_manage_quick:n	31, 83, 319	
L		
\l_keys_current_choice_tlp	46, 51	
\l_niib_after_cite_bool		
..... 27, 64, 68, 72, 191, 205, 216		
\l_niib_before_cite_bool		
..... 27, 65, 69, 73, 187, 202, 213		
\l_niib_err_option_tlp	6, 7, 14	
\l_niib_field_tlp	54, 135	
\l_niib_key_after_tlp	54, 206	
\l_niib_key_before_tlp	54, 203	
\l_niib_key_mixed_tlp	54, 208	
\l_niib_keyname_tlp ...	138, 254, 261, 269	
\l_niib_keyword_tlp	54, 142	
\l_niib_name_tlp	19, 54	
\l_niib_package_option_tlp	5	
\l_niib_placement_tlp	61	
\l_niib_prefix_tlp	75, 122, 276, 281	
\l_niib_presort_after_tlp	75, 217	
\l_niib_presort_before_tlp	75, 214	
\l_niib_presort_mixed_tlp	75, 219	
\l_niib_record_tlp	75, 133	
\l_niib_write_key_bool	81, 137	
\l_tmpa_bool	340, 343, 352, 359	
\let	377	
\let:Nc	45, 50	
\let:NN	42,	
225, 226, 227, 231, 232, 233, 237, 238,		
239, 243, 244, 245, 286, 289, 290, 304, 307		
\let_new:NN	29, 30, 263,	
270, 301, 311, 312, 313, 314, 315, 316, 365		
M		
\MessageBreak	129	
N		
name (option)	3	
\NewDocumentCommand	99,	
109, 117, 149, 169, 174, 178, 182, 280		
\newif	283	
\niib_after_cite:n	194, 255	
\niib_auxhandle:	284	
\niib_bibliography:n	255	
\niib_cite:n	31, 196, 266, 292, 306	
\niib_convert_endnotes:	29, 45, 317	
\niib_convert_endnotes_false: ..	223, 322	
\niib_convert_endnotes_true: ..	223, 321	
\niib_convert_footnotes:	29, 50, 318	
\niib_convert_footnotes_false: .	223, 324	
\niib_convert_footnotes_true: ..	223, 323	
\niib_debug:n	34,	
128, 186, 188, 192, 224, 230, 236, 242		
\niib_key:	139, 201	
\niib_notemark:n	112, 114, 185	
\niib_prefix	262	
\niib_presort:	144, 201	
\niib_rerun_check:n	331, 380, 381	
\niib_rerun_log: ...	334, 338, 360, 366, 387	
\niib_rerun_test:n .	341, 349, 350, 358, 365	
\niib_saved_endnote:	225, 311	
\niib_saved_endnotemark:	226, 311	
\niib_saved_endnotetext:	227, 311	
\niib_saved_footnote:	237, 311	
\niib_saved_footnotemark: ..	238, 248, 311	
\niib_saved_footnotetext: ..	239, 251, 311	
\niib_thenote:	18, 103, 112, 117	
\niibaftercite	163, 174	
\niibbeforecite	153, 174	

<code>\niibsetup</code>	2, 9, 169	<code>\ProcessOptions</code>	16
<code>\niibtotallnotes</code>	182, 384	<code>\protect</code>	251
<code>\nocite</code>	160, 176	<code>\protected@xdef</code>	249
<code>\num_compare:nNnF</code>	21, 386	<code>\ProvidesExplPackage</code>	2
<code>\num_compare:nNnT</code>	20		
		R	
		<code>record (option)</code>	3
		<code>\RequirePackage</code>	4
		S	
		<code>\scan_stop:</code>	16, 29, 30, 290, 301, 377
		T	
		<code>\text_put_four_sp:</code>	38
		<code>\text_put_sp:</code>	9, 135, 138, 141, 143, 257, 258, 327, 328, 368, 371
		<code>\thanks</code>	247
		<code>\the</code>	251
		<code>\tlp_gput_right:Nn</code>	173, 176, 294
		<code>\tlp_gset:Nn</code>	269
		<code>\tlp_new:N</code>	172, 254
		<code>\tlp_new:Nn</code>	6
		<code>\tlp_put_right:Nn</code>	296
		<code>\tlp_set:Nn</code>	261
		<code>\tlp_use:N</code>	203, 206, 208, 214, 217, 219
		<code>\token_to_str:N</code> ..	8, 153, 163, 257, 327, 384
		U	
		<code>\use_none:n</code>	42
		<code>\use_none_i:n</code>	365
		W	
		<code>write key field (option)</code>	3
O			
options:			
<code>cite</code>	3		
<code>convert endnotes</code>	3		
<code>convert footnotes</code>	3		
<code>debug</code>	3		
<code>field</code>	3		
<code>key after</code>	3		
<code>key before</code>	3		
<code>key mixed</code>	3		
<code>name</code>	3		
<code>placement</code>	3		
<code>prefix</code>	3		
<code>presort after</code>	3		
<code>presort before</code>	3		
<code>presort mixed</code>	3		
<code>record</code>	3		
<code>write key field</code>	3		
P			
<code>placement (option)</code>	3		
<code>prefix (option)</code>	3		
<code>presort after (option)</code>	3		
<code>presort before (option)</code>	3		
<code>presort mixed (option)</code>	3		
<code>\printbibliography</code>	263		
<code>\printbibnotes</code>	255		