

xnotes2bib — Integrating notes into the bibliography*

Joseph Wright[†]

Released 2008/09/09

Abstract

The xnotes2bib package is an *experimental* re-implementation of notes2bib using L^AT_EX3 commands internally (taken from the expl3 bundle). The user interface is almost identical to notes2bib.

The xnotes2bibpackage 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 L^AT_EX citations. Both sorted and unsorted bibliography styles are supported.

Contents		5 Implementation	4
1 Introduction	1	5.1 Preliminaries	4
2 Using the package	2	5.2 User options	5
2.1 Macros	2	5.3 User macros	8
2.2 Control values	2	5.4 Macros for the .aux file . . .	10
3 Testing the package	3	5.5 Internal macros	11
4 Installation	4	5.6 Finalising	13

1 Introduction

The new code syntax introduced by L^AT_EX3 is very different from plain T_EX or L^AT_EX2_ε. This means that converting packages to L^AT_EX3 requires working from the ground up if

*This file describes version v0.0, last revised 2008/09/09.

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

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 ϵ 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>	To convert endnotes and footnotes into bibnotes, the options <code>convert endnotes</code> and <code>convert footnotes</code>
<code>convert footnotes</code>	

	convert	footnotes	are available. Both taken Boolean values, and are set to false when loading <code>xnotes2bib</code> . If endnotes are not defined elsewhere, the <code>convert</code> <code>endnotes</code> only defines them when set to true.
placement			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.
key after			Most sorted bibliography styles provide methods for controlling the sort order of entries. With standard <code>BIBTEX</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 key 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>ml</code> and <code>mm</code> , respectively. The <code>presort</code> system is described in the <code>biblatex</code> manual.
key before			
key mixed			
presort after			
presort before			
presort mixed			
cite			As <code>xnotes2bib</code> uses <code>BIBTEX</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> .
field			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>BIBTEX</code> record type used. The standard values are <code>note</code> and <code>Misc</code> , respectively. Finally, the <code>notename</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>BIBTEX</code> .
notename			
prefix			
record			
debug			The <code>debug</code> Boolean option is available for understanding what the package is doing. When set to true, 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 `LATEX3` ideas are encouraged to read the source and make suggestions. The new coding environment requires a more structured approach than `LATEX2ε`, 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 \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 \TeX Live or `initexmf -u` for $\text{MiK}\TeX$)

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 \NeedsTeXFormat{LaTeX2e}
2 \begingroup
3   \@ifundefined{eTeXversion}{%
4     \PackageError{xnotes2bib}
5       {Not running under e-TeX}
6       {This package requires e-TeX: enable the extensions!}%
7   \endgroup\endinput
8 }{%
9   \endgroup
10 }
11 \RequirePackage{l3names}
12 \ProvidesExplPackage
13   {xnotes2bib}{2008/09/09}{0.0}
```

```

14 {Integrating notes into the bibliography (expl3 version)}
15 \RequirePackage{expl3,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 $\text{\LaTeX}2_{\epsilon}$ macros should still be valid in $\text{\LaTeX}3$. There are also a few @ macros where things are tidied closely to the $\text{\LaTeX}2_{\epsilon}$ kernel at present (for example `\@files`).

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

```

16 \DeclareOption*{
17   \PackageWarning{xnotes2bib}{
18     All~package~control~should~be~given~using~the~\MessageBreak
19     \token_to_string:N\niibsetup\space macro
20   }
21 }
22 \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).

```

23 \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.

```

24 \def_new:Npn \niib_thenote: {
25   \l_niib_notename_tlp
26   \num_compare:nNtT {\g_niib_totalnotes_int} > {\c_nine}{
27     \num_compare:nNfT {\g_niib_note_int} > {\c_nine}{
28       0
29     }
30   }
31   \int_to_arabic:n \g_niib_note_int
32 }

```

5.2 User options

`\g_niib_KV_bool_clist` To be going on with, the key-defining macros are created using `keyval`.

```

\niib_KV_bool:nT
\niib_KV_choice:nnT
\niib_KV_cs:n:n
\niib_KV_tlp:n
33 \RequirePackage{keyval}
34 \clist_new:N \g_niib_KV_bool_clist
35 \clist_put_left:Nn \g_niib_KV_bool_clist {true,false}
36 \def_new:Npn \niib_KV_bool:nT #1#2 {
37   \define@key{niib}{#1}[true]{
38     \clist_if_in:cnTF{\g_niib_KV_bool_clist}{##1}{

```

```

39     #2
40   }{
41     \PackageError{xnotes2bib}{
42       Unknown~choice~'##1'
43     }{
44       The~Boolean~option~'##1'~accepts~only~the~\MessageBreak
45       values~true~and~false
46     }
47   }
48 }
49 }
50 \def_new:Npn\niib_KV_choice:nnT #1#2#3 {
51   \cs_if_free:cTF{g_niib_KV_#1_clist}{
52     \clist_new:c{g_niib_KV_#1_clist}
53   }{
54     \clist_gclear:c{g_niib_KV_#1_clist}
55   }
56   \clist_put_left:cn{g_niib_KV_#1_clist}{#2}
57   \define@key{niib}{#1}{
58     \clist_if_in:cnTF{g_niib_KV_#1_clist}{##1}{
59       #3
60     }{
61       \PackageError{xnotes2bib}{
62         Unknown~choice~'##1'
63       }{
64         The~option~'##1'~accepts~only~the~values:\MessageBreak
65         #2
66       }
67     }
68   }
69 }
70 \def_new:Npn\niib_KV_cs_n:n #1 {
71   \define@key{niib}{#1}{
72     \def:cNn{niib_#1:n} 0 {##1}
73   }
74 }
75 \def_new:Npn\niib_KV_tlp:n #1 {
76   \tlp_new:c {l_niib_#1_tlp}
77   \define@key{niib}{#1}{
78     \tlp_set:cn {l_niib_#1_tlp}{##1}
79   }
80 }

```

\niib_cite:n Options for storing macro names

```
81 \niib_KV_cs_n:n{cite}
```

\l_niib_field_tlp A series of value-storing are created.

```

\l_niib_key after_tlp
\l_niib_key before_tlp
\l_niib_key mixed_tlp
\l_niib_notename_tlp
\l_niib_prefix_tlp
\l_niib_presort after_tlp
\l_niib_presort before_tlp
\l_niib_presort mixed_tlp
\l_niib_record_tlp

```

```

82 \niib_KV_tlp:n{field}
83 \niib_KV_tlp:n{key~after}
84 \niib_KV_tlp:n{key~before}
85 \niib_KV_tlp:n{key~mixed}
86 \niib_KV_tlp:n{notename}
87 \niib_KV_tlp:n{prefix}
88 \niib_KV_tlp:n{presort~after}
89 \niib_KV_tlp:n{presort~before}
90 \niib_KV_tlp:n{presort~mixed}
91 \niib_KV_tlp:n{record}

```

\niib_convert_endnotes: None of the Boolean keys actually work quite as simply as creating a switch. For footnote and endnote manipulation, the keys created here have to be altered later in any case.

```

92 \let_new:NN\niib_convert_endnotes:\scan_stop:
93 \let_new:NN\niib_convert_footnotes:\scan_stop:
94 \niib_KV_bool:nT{convert~endnotes}{
95   \let:Nc\niib_convert_endnotes:{niib_convert_endnotes_#1:}
96 }
97 \niib_KV_bool:nT{convert~footnotes}{
98   \let:Nc\niib_convert_footnotes:{niib_convert_footnotes_#1:}
99 }

```

\niib_debug:n The debug option is implemented as a choice, to save on some complexity.

```

\niib_debug_true:n
\niib_debug_false:n
100 \niib_KV_bool:nT{debug}{
101   \let:Nc\niib_debug:n{niib_debug_#1:n}
102 }
103 \def_new:Npn\niib_debug_true:n #1 {
104   \PackageInfo{xnotes2bib}{#1}
105 }
106 \let_new:NN\niib_debug_false:n\use_none:n

```

\l_niib_after_cite_bool The placement option sets one or other flag if needed.

```

\l_niib_before_cite_bool
107 \bool_new:N\l_niib_after_cite_bool
108 \bool_new:N\l_niib_before_cite_bool
109 \niib_KV_choice:nnT{placement}{after,before,mixed}{
110   \bool_set_false:N\l_niib_before_cite_bool
111   \bool_set_false:N\l_niib_after_cite_bool
112   \cs_if_really_exist:cT{l_niib_#1_cite_bool}{
113     \bool_set_true:c{l_niib_#1_cite_bool}
114   }
115 }

```

All of the default values are set.

```

116 \setkeys{niib}{

```

```

117 cite          = \cite,
118 debug          = false,
119 field          = note,
120 key~after      = zzz,
121 key~before     = aaa,
122 key~mixed      = {},
123 notename       = Bibnote,
124 placement      = mixed,
125 prefix         = niib-,
126 presort~after  = mn,
127 presort~before = ml,
128 presort~mixed  = mm,
129 record         = Misc}

```

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 \LaTeX 3, it is assumed $\epsilon\text{-TeX}$ is available and so `\scantokens` can be used on the mandatory argument. Hence everything is absorbed here without worrying about category codes.

```

130 \NewDocumentCommand{\bibnote}{o>{P}m}{
131   \IfNoValueTF{#1}{
132     \int_gincr:N \g_niib_note_int
133     \bibnotetext{#2}
134     \bibnotemark[\niib_thenote:]
135   }{
136     \bibnotetext[#1]{#2}
137     \bibnotemark[#1]
138   }
139 }

```

`\bibnotemark` [#1]: label

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

```

140 \NewDocumentCommand{\bibnotemark}{o}{
141   \IfNoValueTF{#1}{
142     \int_gincr:N \g_niib_note_int
143     \niib_notemark:n{\niib_thenote:}
144   }{

```



```

145 \niib_notemark:n{#1}
146 }
147 }

```

`\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 $\text{\LaTeX} 2_{\epsilon}$ flag (and so the tradition syntax).

```

148 \NewDocumentCommand{\bibnotetext}{0{\niib_thenote:}>{P}m}{
149 \if@files

```

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

```

150 \cs_if_free:NT \g_niib_out_stream {
151 \iow_new:N \g_niib_out_stream
152 \iow_open:Nn \g_niib_out_stream
153 {\l_niib_prefix_tlp\c_job_name_tlp.bib}
154 \iow_expanded:Nn \g_niib_out_stream
155 {This-is-an-auxiliary-file-used-by-the-‘xnotes2bib’~
156 package.^J This-file-may-safely-be-deleted.^J It~
157 will-be-recreated-as-required.^J}
158 }

```

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.

```

159 \niib_debug:n{
160 Writing~text:\MessageBreak #2\MessageBreak to~database~
161 for~note~‘#1’
162 }
163 \iow_expanded:Nn \g_niib_out_stream {
164 @\l_niib_record_tlp{
165 #1,^J
166 \space\space \l_niib_field_tlp\space
167 =~{\exp_not:n{#2}},^J
168 \space\space \l_niib_keyname_tlp\space
169 =~{\niib_key:#1},^J
170 \space\space presort~
171 =~{\niib_presort:},^J
172 }^J
173 }
174 \fi
175 }

```

`\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.

```

176 \NewDocumentCommand{\flushnotestack}{}{
177   \clist_if_empty:NF\g_niib_before_cite_clist{
178     \if@files
179       \iow_expanded:Nn\@auxout{
180         \token_to_string:N\niibbeforecite{
181           \g_niib_before_cite_clist
182         }
183       }
184     \fi
185   }
186   \clist_if_empty:NF\g_niib_after_cite_clist{
187     \nocite{\g_niib_after_cite_clist}
188     \if@files
189       \iow_expanded:Nn\@auxout{
190         \token_to_string:N\niibaftercite{\g_niib_after_cite_clist}
191       }
192     \fi
193   }
194   \clist_gclear:N\g_niib_after_cite_clist
195 }

```

`\niibsetup`

```

196 \NewDocumentCommand{\niibsetup}{m}{\setkeys{niib}{#1}}

```

5.4 Macros for the .aux file

To allow tracking of the need to re-run L^AT_EX, 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`

```

197 \clist_new:N\g_niib_before_cite_old_clist
198 \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.

```

199 \tlp_new:N\g_niib_document_tlp
200 \tlp_put_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.

```
201 \NewDocumentCommand{\niibbeforecite}{m}{
202   \clist_gput_right:Nn\g_niib_before_cite_old_clist{#1}
203   \tlp_gput_right:Nn\g_niib_document_tlp{\nocite{#1}}
204 }
205 \NewDocumentCommand{\niibaftercite}{m}{
206   \clist_gput_right:Nn\g_niib_after_cite_old_clist{#1}
207 }
```

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

```
208 \int_new:N \g_niib_totalnotes_int
```

`\niibttotalnotes` #1: number

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

```
209 \NewDocumentCommand{\niibttotalnotes}{m}{
210   \int_gset:Nn \g_niib_totalnotes_int {#1}
211 }
```

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.

```
212 \def_new:Npn \niib_notemark:n #1 {
213   \niib_debug:n{Creating~citation~for~note~‘#1’}
214   \bool_if:NT\l_niib_before_cite_bool{
215     \niib_debug:n{Note~‘#1’~before~real~citations}
216     \clist_gput_right:Nx\g_niib_before_cite_clist{#1}
217   }
218   \bool_if:NTF\l_niib_after_cite_bool{
219     \niib_debug:n{Note~‘#1’~after~real~citations}
220     \clist_gput_right:Nx\g_niib_after_cite_clist{#1}
221     \niib_after_cite:n{#1}
222   }{
223     \niib_cite:n{#1}
224   }
225 }
```

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

`\g_niib_after_cite_clist`

```
226 \clist_new:N\g_niib_before_cite_clist
227 \clist_new:N\g_niib_after_cite_clist
```

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

```

228 \def_new:Npn\niib_key: {
229   \bool_if:NTF\l_niib_before_cite_bool{
230     \tlp_use:c{l_niib_key~before_tlp}
231   }{
232     \bool_if:NTF\l_niib_after_cite_bool{
233       \tlp_use:c{l_niib_key~after_tlp}
234     }{
235       \tlp_use:c{l_niib_key~mixed_tlp}
236     }
237   }
238 }
239 \def_new:Npn\niib_presort: {
240   \bool_if:NTF\l_niib_before_cite_bool{
241     \tlp_use:c{l_niib_presort~before_tlp}
242   }{
243     \bool_if:NTF\l_niib_after_cite_bool{
244       \tlp_use:c{l_niib_presort~after_tlp}
245     }{
246       \tlp_use:c{l_niib_presort~mixed_tlp}
247     }
248   }
249 }

```

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

```

\niib_convert_endnotes_true:
\niib_convert_footnotes_false:
\niib_convert_footnotes_true:
250 \def_new:Npn\niib_convert_endnotes_false: {
251   \niib_debug:n{Restoring~normal~endnotes}
252   \let:NN\endnote\niib_saved_endnote:
253   \let:NN\endnotemark\niib_saved_endnotemark:
254   \let:NN\endnotetext\niib_saved_endnotetext:
255 }
256 \def_new:Npn\niib_convert_endnotes_true: {
257   \niib_debug:n{Converting~endnotes~to~bibnotes}
258   \let:NN\endnote\bibnote
259   \let:NN\endnotemark\bibnotemark
260   \let:NN\endnotetext\bibnotetext
261 }
262 \def_new:Npn\niib_convert_footnotes_false: {
263   \niib_debug:n{Restoring~normal~footnotes}
264   \let:NN\footnote\niib_saved_footnote:
265   \let:NN\footnotemark\niib_saved_footnotemark:
266   \let:NN\footnotetext\niib_saved_footnotetext:
267 }
268 \def_new:Npn\niib_convert_footnotes_true: {
269   \niib_debug:n{Converting~footnotes~to~bibnotes}
270   \let:NN\footnote\bibnote

```

```

271 \let:NN\footnotemark\bibnotemark
272 \let:NN\footnotetext\bibnotetext
273 }

```

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

```

274 \def:Npn\thanks #1 {
275   \niib_saved_footnotemark:
276   \protected@xdef\@thanks{\@thanks
277   \protect\niib_saved_footnotetext:[\the\c@footnote]{#1}}
278 }

```

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.

```

279 \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 `\@files` switch to prevent writing at the wrong time.

```

280 \AtBeginDocument{
281   \PackageInfo{xnotes2bib}{
282     Beginning~\token_to_string:N\AtBeginDocument\space tasks
283   }
284   \@ifpackageloaded{biblatex}{
285     \tlp_set:Nn \l_niib_keyname_tlp {keysort}
286     \gappto\blx@bibfiles{,\niib_prefix\c_job_name_tlp}
287     \let_new:NN\printbibnotes\printbibliography
288     \def_new:Npn\niib_after_cite:n #1 {
289       \AtNextCite{\@filesfalse}
290       \niib_cite:n{#1}
291     }
292   }{
293     \tlp_gset:Nn \l_niib_keyname_tlp {key}
294     \let_new:NN\niib_bibliography:n\bibliography
295     \def:Npn\bibliography #1 {
296       \int_compare:nNnTF{\g_niib_note_int}={\c_zero}{
297         \niib_bibliography:n{#1}
298       }{
299         \niib_bibliography:n{

```

```

300         #1,\l_niib_prefix_tlp\c_job_name_tlp
301     }
302 }
303 }
304 \NewDocumentCommand{\printbibnotes}{}{
305     \niib_bibliography:n{\l_niib_prefix_tlp\c_job_name_tlp}
306 }

```

\ifniib@filesw There is a need to back-up the \@filesw switch. As this is a L^AT_EX 2_ε switch, the name and creation method used here are not L^AT_EX 3-based.

```

307     \newif\ifniib@filesw

```

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

```

\niib_auxhandle:
308     \@ifpackageloaded{cite}{
309         \def_new:Npn\niib_after_cite:n #1 {
310             \let:NN \ifniib@filesw \if@filesw
311             \@fileswfalse
312             \def:Npn \niib_auxhandle: {
313                 \let:NN \if@filesw \ifniib@filesw
314                 \let:NN \niib_auxhandle: \scan_stop:
315             }
316             \niib_cite:n{#1}
317         }
318         \tlp_gput_right:Nn \g_niib_document_tlp {
319             \cs_if_exist:NTF\@restore@auxhandle{
320                 \tlp_put_right:Nn\@restore@auxhandle{\niib_auxhandle:}
321             }{
322                 \def_new:Npn\@restore@auxhandle {\niib_auxhandle:}
323             }
324         }
325         \let_new:NN\niib_auxhandle:\scan_stop:

```

Life is slightly easier without cite.

```

326     }{
327         \def_new:Npn\niib_after_cite:n #1 {
328             \let\ifniib@filesw\if@filesw
329             \@fileswfalse
330             \niib_cite:n{#1}
331             \let\if@filesw\ifniib@filesw
332         }
333     }
334 }

```

\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_footnotemark:
\niib_saved_footnotetext:
\niib_saved_endnote:
\niib_saved_endnotemark:
\niib_saved_endnotetext:

```

```

335 \let_new:NN\niib_saved_footnote:\footnote
336 \let_new:NN\niib_saved_footnotemark:\footnotemark
337 \let_new:NN\niib_saved_footnotetext:\footnotetext
338 \let_new:NN\niib_saved_endnote:\endnote
339 \let_new:NN\niib_saved_endnotemark:\endnotemark
340 \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.

```

341 \niib_convert_endnotes:
342 \niib_convert_footnotes:
343 \niib_KV_bool:nT{convert~endnotes}{
344   \use:c{niib_convert_endnotes_#1:}
345 }
346 \niib_KV_bool:nT{convert~footnotes}{
347   \use:c{niib_convert_footnotes_#1:}
348 }
349 \PackageInfo{xnotes2bib}{
350   Finished~\token_to_string:N\AtBeginDocument\space tasks
351 }
352 }

```

`\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.

```

353 \def_new:Npn\niib_rerun_check:n #1 {
354   \clist_if_empty:cTF {g_niib_old#1cite_clist}{
355     \clist_if_empty:cF {g_niib_#1cite_clist}{
356       \niib_rerun_log:
357     }
358   }{
359     \clist_if_empty:cTF {g_niib_#1cite_clist}{
360       \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.

```

361   }{
362     \bool_set_false:N\l_tmpa_bool
363     \def:Npn\niib_rerun_test:n ##1 {
364       \clist_if_in:cnTF{g_niib_old#1cite_clist}{##1}{
365         \bool_set_true:N\l_tmpa_bool
366         \clist_map_break:w
367       }{
368         }
369     }
370     \clist_map_function:cn{g_niib_#1cite_clist}

```

```

371     \niib_rerun_test:n
372     \def:Npn\niib_rerun_test:n ##1 {
373         \clist_if_in:cnTF{g_niib_#1cite_clist}{##1}{
374             \bool_set_true:N\l_tmpa_bool
375             \clist_map_break:w
376         }{
377         }
378     }
379     \clist_map_function:cn{g_niib_old#1cite_clist}
380     \niib_rerun_test:n
381     \bool_if:NF\l_tmpa_bool{
382         \niib_rerun_log:
383     }
384 }
385 }
386 }

```

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

```

387 \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.

```

388 \def_new:Npn\niib_rerun_log: {
389     \PackageInfo{xnotes2bib}{
390         To~get~notes~in~the~correct~order,~please~\MessageBreak
391         rerun~LaTeX,~(re)run~BibTeX~on~the~file~\c_job_name_tlp.aux
392         \MessageBreak and~rerun~LaTeX~again~afterwards
393     }
394     \io_put_log:x {
395         REQ:3:latex:REQ\iow_newline:
396         REQ:2:bibtex:REQ\iow_newline:
397         REQ:1:latex:REQ\iow_newline:
398     }
399     \let\niib_rerun_log:\scan_stop:
400 }

```

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

```

401 \AtEndDocument{
402     \niib_rerun_check:n{before}
403     \niib_rerun_check:n{after}
404     \flushnotestack
405     \iow_expanded:Nn\@auxout{
406         \token_to_string:N\niibttotalnotes{
407             \int_to_arabic:n \g_niib_note_int
408         }
409     }
410     \num_compare:nNnF{\g_niib_totalnotes_int}={\g_niib_note_int}{

```



```

411   \niib_rerun_log:
412 }
413 \cs_if_free:NF \g_niib_out_stream{
414   \iow_close:N \g_niib_out_stream
415 }
416 }

```

Change History

v0.0		named from \g_niib_oldbeforecite_clist 10
\@restore@auxhandle: Moved patch code to after \AtBeginDocument ...	14	\l_niib_after_cite_bool: Renamed from \l_niib_after_bool 7
General: Replaced \cs_use:c by \use:c	4	\l_niib_before_cite_bool: Renamed from \l_niib_before_bool 7
Replaced use of \def_new:NNn and \def:NNn with \def_new:Npn and \def:Npn, respectively	4	\l_niib_keyname_tlp: Renamed from \niib_keyname: 13
\g_niib_after_cite_clist: Renamed from \g_niib_aftercite_clist ..	11	\niib_after_cite:n: Renamed from \niib_aftercite:n 13
\g_niib_after_cite_old_clist: Re- named from \g_niib_oldaftercite_clist	10	\niib_KV_tlp:n: Renamed and re- coded from \niib_KV_string:n .. 5
\g_niib_before_cite_clist: Re- named from \g_niib_beforecite_clist	11	\niib_notemark:n: Replaced \clist_put_right:Nx by \clist_gput_right:Nx 11
\g_niib_before_cite_old_clist: Re-		\niib_rerun_log:: Maximum of one message per run 16

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 280, 282, 350	\bibnotetext 2,
\@auxout 179, 189, 405	\AtEndDocument 401	133, 136, <u>148</u> , 260, 272
\@fileswfalse 289, 311, 329	\AtNextCite 289	\blx@bibfiles 286
\@ifpackageloaded 284, 308		\bool_if:NF 381
\@ifundefined 3	B	\bool_if:NT 214
\@restore@auxhandle . 308	\begingroup 2	\bool_if:NTF
\@thanks 276	\bibliography 294, 295	218, 229, 232, 240, 243
	\bibnote ... 2, <u>130</u> , 258, 270	\bool_new:N 107, 108
A	\bibnotemark 2,	\bool_set_false:N ...
\AtBeginDocument	134, 137, <u>140</u> , 259, 271 110, 111, 362

\bool_set_true:c	113	256, 262, 268, 288,	\int_gincr:N	132, 142
\bool_set_true:N .	365, 374	309, 322, 327, 353, 388	\int_gset:Nn	210
C		\define@key .	37, 57, 71, 77	\int_new:N
\c@footnote	277	\document	200	\int_to_arabic:n . .
\c_job_name_tlp		E		
	153, 286, 300, 305, 391	\endgroup	7, 9	\io_put_log:x
\c_nine	26, 27	\endinput	7	\iow_close:N
\c_zero	296	\endnote	252, 258, 338	\iow_expanded:Nn
\cite	117	\endnotemark .	253, 259, 339	154, 163, 179, 189, 405
cite (option)	3	\endnotetext .	254, 260, 340	\iow_new:N
\clist_gclear:c	54	\exp_not:n	167	151
\clist_gclear:N	194	K		
\clist_gput_right:Nn .		key after (option) 3		
	202, 206	key before (option) 3		
\clist_gput_right:Nx .		key mixed (option) 3		
	216, 220	L		
\clist_if_empty:cF . .	355	\fi	174, 184, 192	\l_niib_after_cite_bool
\clist_if_empty:cTF .		field (option)	3 107, 218, 232, 243
	354, 359	\flushnotestack .	2, 176, 404	\l_niib_before_cite_bool
\clist_if_empty:NF	177, 186	\footnote . . .	264, 270, 335 107, 214, 229, 240
\clist_if_in:cnTF . .		\footnotemark .	265, 271, 336	\l_niib_field_tlp .
	38, 58, 364, 373	\footnotetext .	266, 272, 337	82, 166
\clist_map_break:w	366, 375	G		
\clist_map_function:cN		\g_niib_after_cite_clist		\l_niib_key_ after_tlp .
	370, 379		186,	82
\clist_new:c	52		187, 190, 194, 220, 226	\l_niib_key_ before_tlp .
\clist_new:N		\g_niib_after_cite_old_clist		82
	34, 197, 198, 226, 227		197, 206	\l_niib_key_ mixed_tlp .
\clist_put_left:cn . .	56	\g_niib_before_cite_clist		82
\clist_put_left:Nn . .	35		177, 181, 216, 226	\l_niib_keyname_tlp .
convert endnotes (op-		\g_niib_before_cite_old_clist	 168, 279, 285, 293
tion)	2		197, 202	\l_niib_notename_tlp .
convert footnotes (op-		\g_niib_document_tlp 25, 82
tion)	2		199, 203, 318	\l_niib_prefix_tlp . .
\cs_if_exist:NTF . . .	319	\g_niib_KV_bool_clist .	33 82, 153, 300, 305
\cs_if_free:cTF	51	\g_niib_note_int		\l_niib_presort_ after_tlp
\cs_if_free:NF	413		23, 27, 31, 82
\cs_if_free:NT	150		132, 142, 296, 407, 410	\l_niib_presort_ before_tlp
\cs_if_really_exist:cT	112	\g_niib_out_stream 82
D			150, 413, 414	\l_niib_presort_ mixed_tlp
debug (option)	3	\g_niib_totalnotes_int	 82
\DeclareOption	16		26, 208, 210, 410	\l_niib_record_tlp .
\def:cNn	72	\gappto	286	82, 164
\def:Npn		I		
	274, 295, 312, 363, 372	\if@filesw . . .	149, 178,	\l_tmpa_bool
\def_new:Npn	24,		188, 310, 313, 328, 331 362, 365, 374, 381
	36, 50, 70, 75, 103,	\ifniib@filesw		\let
	212, 228, 239, 250,		307, 310, 313, 328, 331	328, 331, 399
		\IfNoValueTF	131, 141	\let:Nc
		\int_compare:nNnTF . .	296	95, 98, 101
				\let:NN
				254, 258, 259, 260,
				264, 265, 266, 270,
				271, 272, 310, 313, 314
				\let_new:NN
			 92, 93, 106, 287,

294, 325, 335, 336, 337, 338, 339, 340, 387			
M			
\MessageBreak			
18, 44, 64, 160, 390, 392			
N			
\NeedsTeXFormat	1		
\NewDocumentCommand .			
130, 140, 148, 176, 196, 201, 205, 209, 304			
\newif	307		
\niib_after_cite:n	221, 280		
\niib_auxhandle:	308		
\niib_bibliography:n .	280		
\niib_cite:n	81, 223, 290, 316, 330		
\niib_convert_endnotes:	92, 341		
\niib_convert_endnotes_false:	250		
\niib_convert_endnotes_true:	250		
\niib_convert_footnotes:	92, 342		
\niib_convert_footnotes_false:	250		
\niib_convert_footnotes_true:	250		
\niib_debug:n	100, 159, 213, 215, 219, 251, 257, 263, 269		
\niib_debug_false:n .	100		
\niib_debug_true:n ..	100		
\niib_key:	169, 228		
\niib_KV_bool:nT	33, 94, 97, 100, 343, 346		
\niib_KV_choice:nnT	33, 109		
\niib_KV_cs_n:n ...	33, 81		
\niib_KV_tlp:n	33, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91		
\niib_notemark:n	143, 145, 212		
\niib_prefix	286		
\niib_presort: ...	171, 228		
\niib_rerun_check:n .	353, 402, 403		
\niib_rerun_log:	356, 360, 382, 388, 411		
\niib_rerun_test:n ..	363, 371, 372, 380, 387		
\niib_saved_endnote: .	252, 335		
\niib_saved_endnotemark:	253, 335		
\niib_saved_endnotetext:	254, 335		
\niib_saved_footnote:	264, 335		
\niib_saved_footnotemark:	265, 275, 335		
\niib_saved_footnotetext:	266, 277, 335		
\niib_thenote:	24, 134, 143, 148		
\niibaftercite ...	190, 201		
\niibbeforecite ..	180, 201		
\niibsetup	2, 19, 196		
\niibtotallnotes ..	209, 406		
\nocite	187, 203		
notename (option)	3		
\num_compare:nNnF .	27, 410		
\num_compare:nNnT	26		
O			
options:			
cite	3		
convert endnotes ...	2		
convert footnotes ..	2		
debug	3		
field	3		
key after	3		
key before	3		
key mixed	3		
notename	3		
placement	3		
prefix	3		
presort after	3		
presort before	3		
presort mixed	3		
record	3		
P			
\PackageError	4, 41, 61		
\PackageInfo	104, 281, 349, 389		
\PackageWarning	17		
placement (option)	3		
prefix (option)	3		
presort after (option) ..	3		
presort before (option) .	3		
presort mixed (option) ..	3		
\printbibliography ..	287		
\printbibnotes	280		
\ProcessOptions	22		
\protect	277		
\protected@xdef	276		
\ProvidesExplPackage ..	12		
R			
record (option)	3		
\RequirePackage .	11, 15, 33		
S			
\scan_stop:	22, 92, 93, 314, 325, 399		
\setkeys	116, 196		
\space	19,		
166, 168, 170, 282, 350			
T			
\thanks	274		
\the	277		
\tlp_gput_right:Nn	203, 318		
\tlp_gset:Nn	293		
\tlp_new:c	76		
\tlp_new:N	199, 279		
\tlp_put_right:Nn	200, 320		
\tlp_set:cn	78		
\tlp_set:Nn	285		
\tlp_use:c	230,		
233, 235, 241, 244, 246			
\token_to_string:N	19,		
180, 190, 282, 350, 406			
U			
\use:c	344, 347		
\use_none:n	106		
\use_none_i:n	387		