

# The `latex-lab-bib` package

## Changes and additions to the kernel related to tagging and links in citations and bibliography entries

L<sup>A</sup>T<sub>E</sub>X Project\*

v0.81c 2025-05-09

### Abstract

## 1 Introduction

The followings contains small changes to improve tagging of bibliography entries and citations.

The tagging of the standard bibliography is actually quite straightforward: A bibliography is typically a list with a heading and the code which tags sectioning commands and lists handles that.

There are here only two problems:

- The structure number of the LI element created by a `\bibitem` must be recorded somehow to allow to reference it in a `\cite`.
- `hyperref` redefines the item command and so breaks the list structure see <https://github.com/latex3/latex2e/discussions/1010#discussioncomment-5565418>

Both problems are rather easy to resolve, but it must be checked if other packages interfere again by redefining the commands.

More difficult is the tagging of citation commands. Citations should be inside a Reference structure and contain a `/Ref` entry pointing to the relevant item in the bibliography. For simple citations like “[1]” or “Doody (2023)” this is easy, but it is not obvious how to handle combined citations like “Doody (2003,2018)” (or even compressed citations like “[1-3]”). The current implementation follows the links: whatever `hyperref` would link is set as the reference.

There exist various packages which over the years tried to improve and extend the bibliography commands. We discuss here three: `natbib`, `chapterbib` and `biblatex`.

**natbib** It is rather easy to support `natbib`: it has hooks for links and the tagging code can follow. Only a bit coordination with `hyperref` is needed to avoid that `hyperref` remove the tagging code again.

---

\*Initial implementation done by Ulrike Fischer

**chapterbib** In standard LaTeX every bib entry has an unique label which points to the (mandatory and unique) bibliography and the target created by hyperref has the simple form `cite.<key>`. If a package that support multiple bibliographies is used (e.g. chapterbib) this is no longer works: a bib entry `doody` can in one chapter get the label “[1]” and in the other “[5]” or even “Doo19” and naturally links should jump to the relevant chapter bibliography. chapterbib solves this by creating bib keys with a suffix: when reading the `.aux` files it will create the keys `doody@-1` and `doody@-2` where the number is related to the chapter/include, and in the document and in the document `\cite{doody}` will look for `doody@-1` and `doody@-2` depending on the number of the current include. For some unknown reason chapterbib uses two commands to handle the suffix: the command `\@extra@binfo` is written to the aux-files and used when processing the `\bibcite` commands, but in the document `\@extra@b@citeb` is used. Supporting this is straightforward: one only has to take care that the tagging code uses `\@extra@b@citeb` in the relevant places too.

**biblatex** biblatex supports multiple bibliographies out-of-the-box. It numbers the link target by refsection and uses then the name `\the\c@refsection @<key>`.

Printing a bibliography is not required, in this case you get an engine warning and links jump to the begin of the document:

```
name{cite.0@doody} has been referenced but does not exist
```

Bibliographies can be printed more than once by refsection. To avoid duplicated target, biblatex stores the names of the targets in a list and if later it detects that a target name has already been used in a bibliography no new target is created for this item. This means a citation will normally jump to the first bibliography which shows the entry.

The tagging code has to mimic this code. This means that it can’t label every item, but has to test if this anchor is already known.

## 2 Provided or redefined commands

---

<code>\@extra@binfo</code> <code>\@extra@b@citeb</code>	These are taken from hyperref, they are for chapterbib compatibility (and also signal to chapterbib not to change the citation commands)
--	--

---



---

<code>\@bibitem</code> <code>\@lbibitem</code>	The internal item commands.
---	-----------------------------

---

## 3 Implementation

```

1 <*package>
2 <@@=tag>

3 \ProvidesExplPackage {latex-lab-testphase-bib} {\ltlabbibdate} {\ltlabbibversion}
4 {Code related to the tagging of bibliography and cite command}

```

We need at least the block tagging code.

```
5 \RequirePackage{latex-lab-testphase-block}
```

At first we suppress the patches from hyperref. This will only work with the next hyperref!

```
6 \def\hyper@nopatch@bib{}
```

**\@extra@binfo** These are taken from hyperref, they are for chapterbib compatibility (and also signal to  
**\@extra@b@citeb** chapterbib not to change the citation commands)

```
7 \providecommand*\@extra@binfo{}
```

```
8 \providecommand*\@extra@b@citeb{}
```

*(End of definition for \@extra@binfo and \@extra@b@citeb. These functions are documented on page 2.)*

---

**\l\_\_tag\_bib\_citekey\_tl** We can't pass the cite key everywhere as argument so we store it:

```
9 \tl_new:N\l__tag_bib_citekey_tl
```

### 3.1 Handling the bibliography

**\@lbibitem** The item command if an optional argument is used.

We only prepend some code.

```
10 \AddToHookWithArguments{cmd/@lbibitem/before}
```

```
11 {
```

we store the target name for the label.

```
12 \tl_set:Nn\l__tag_bib_citekey_tl{#2}
```

```
13 }
```

The target is added at the begin of the paragraph. We give that a label as it perhaps need to be removed by packages

```
14 \AddToHookWithArguments{cmd/@lbibitem/before}[latex-lab-testphase-bib/target]
```

```
15 {
```

```
16 \AddToHookNext{para/begin}
```

```
17 {
```

```
18 \makebox[0pt][r]{\MakeLinkTarget*{cite.#2\@extra@b@citeb}\hspace{\leftmargin}}
```

```
19 }
```

```
20 }
```

we make a copy to be able to reinstate the definition. This is e.g. currently needed with hyperref.

```
21 \let\@kernel@copy@lbibitem\@lbibitem
```

*(End of definition for \@lbibitem. This function is documented on page 2.)*

**\@bibitem** Similar for \@bibitem. TODO: If hyperref is loaded we will get a second target from the refstepcounter, but this is ignored for now.

```

22 \AddToHookWithArguments{cmd/@bibitem/before}
23 {

```

we store the target name for the label.

```

24 \tl_set:Nn\l__tag_bib_citekey_tl{#1}
25 }

```

The target is added at the begin of the paragraph. We give that a label as it perhaps need to be removed by packages

```

26 \AddToHookWithArguments{cmd/@bibitem/before}[latex-lab-testphase-bib/target]
27 {
28 \AddToHookNext{para/begin}
29 {
30 \makebox[0pt][r]{\MakeLinkTarget*{cite.#1\@extra@b@citeb}\hspace{\leftmargin}}
31 }
32 }

```

```

33 \let\@kernel@copy@bibitem\@bibitem

```

*(End of definition for \@bibitem. This function is documented on page 2.)*

TODO The LI-structure should set a label, we redefine the internal command locally for now, but perhaps this need a recipe?

```

34 \AddToHook{env/thebibliography/begin}
35 {
36 \cs_set:Npn \__block_list_item_begin:
37 {
38 \tag_struct_begin:n
39 {
40 tag=\LItag,
41 label= cite.\l__tag_bib_citekey_tl\@extra@b@citeb
42 }
43 }
44 }

```

### 3.2 Handling citation commands

We redefine similar to hyperref the `\bibcite` command to inject link and structure. Even if it looks a bit odd it is now used for many years and so hopefully compatible with various packages. But differently to hyperref we use the new hooks with arguments. TODO: consider hook name.

```

45 \NewMirroredHookPairWithArguments{bibcite/before}{bibcite/after}{2}
46 \def\bibcite#1#2{%
47 \@newl@bel{b}{#1\@extra@binfo}{%
48 \UseHookWithArguments{bibcite/before}{2}{#1}{#2}
49 #2
50 \UseHookWithArguments{bibcite/after}{2}{#1}{#2}
51 }%
52 }%
53 \let\@kernel@copy@bibcite\bibcite

```

Now we add the tagging structure. TODO: with the next tagpdf version it should no longer be needed to expand the ref key.

```

54 \AddToHookWithArguments{bibcite/before}
55 {
56   \tag_mc_end_push:
57   \exp_args:Ne\tagstructbegin{tag=Reference,ref=cite.#1\@extra@b@citeb}
58   \tagmcbegin{ }
59 }
60 \AddToHookWithArguments{bibcite/after}[tag]
61 {
62   \tag_mc_end:
63   \tagstructend
64   \tag_mc_begin_pop:n{ }
65 }

```

At last the code for hyperref, the link will be inside the reference, but this can be changed with a rule.

```

66 \AddToHook{package/hyperref/after}
67 {
68   \AddToHookWithArguments{bibcite/before}
69   {
70     \hyper@linkstart{cite}{cite.#1\@extra@b@citeb}
71   }
72   \AddToHookWithArguments{bibcite/after}{\hyper@linkend}
73 }

```

### 3.3 Natbib and biblatex support

When hyperref is loaded, both natbib and biblatex use `\hyper@natlinkstart` and `\hyper@natlinkend` to handle the links. We can use the generic hooks to add the tagging code (and the link code from hyperref). We need in part different code for both systems: with biblatex we have to take care that only the first structure sets a label, and if hyperref is not loaded (or deactivated) we will need additional code but this currently doesn't exist. We assume that no document loads both package – that will probably break.

```

74 \newcommand\hyper@natlinkstart[1]{ }
75 \newcommand\hyper@natlinkend{ }

```

With natbib we need to change the hooks to avoid duplicated target as it sets the anchor too. We can not simply empty `\hyper@natanchorstart` as that is used by biblatex.

```

76 \AddToHook{package/natbib/after}
77 {
78   \RemoveFromHook{cmd/@bibitem/before} [latex-lab-testphase-bib/target]
79   \RemoveFromHook{cmd/@lbibitem/before} [latex-lab-testphase-bib/target]
80 }

```

This can be shared by both packages: it will work with natbib with and without hyperref. With biblatex it will work without hyperref as long as `\@extra@b@citeb` is empty

```

81 \AddToHookWithArguments{cmd/hyper@natlinkstart/before}
82 {
83   \leavevmode
84   \tag_mc_end_push:
85   \exp_args:N\tag_struct_begin:n{tag=Reference,ref=cite.#1\@extra@b@citeb}
86   \tag_mc_begin:n{ }
87 }
88 \AddToHook{cmd/hyper@natlinkend/after}
89 {
90   \tag_mc_end:
91   \tag_struct_end:
92   \tag_mc_begin_pop:n{ }
93 }
94

```

if hyperref is loaded we have to revert its definition of the natbib hooks and add its code through the generic hooks. TODO: allow to suppress the natbib code in hyperref.

```

95 \AddToHook{package/hyperref/after}
96 {
97   \renewcommand\hyper@natlinkstart[1]{}
98   \renewcommand\hyper@natlinkend{}
99   \AddToHookWithArguments{cmd/hyper@natlinkstart/before}
100   {
101     \Hy@backout{#1}%

```

natbib passes the \@extra@b@citeb in the argument, and biblatex the refsection, so we only need to add the prefix cite..

```

102     \hyper@linkstart{cite}{cite.#1}%
103     \def\hyper@nat@current{#1}
104   }
105   \AddToHook{cmd/hyper@natlinkend/after}
106   {
107     \hyper@linkend
108   }
109 }
110

```

and now special biblatex code. The list item code has to test if the anchor is already known:

```

111 \AddToHook{cmd/blx@bibinit/after}
112 {
113   \cs_set:Npn \__block_list_item_begin:
114   {
115     \xifinlist{\the\c@refsection @\abx@field@entrykey}{\blx@anchors}
116     {
117       \tag_struct_begin:n
118       {
119         tag=LItag,
120       }
121     }
122     {
123       \tag_struct_begin:n

```

```

124         {
125             tag=\LItag,
126             label= cite.\the\c@refsection @\abx@field@entrykey
127         }
128     }
129 }
130 }

```

biblatex without hyperref is currently not supported but we at least avoid that it errors:

```

131 \AddToHook{package/biblatex/after}
132 {
133     \appto\blx@mknohyperref
134     {
135         \let\blx@anchors\@empty
136         \protected\def\blx@anchor{%
137             \xifinlist{\the\c@refsection @\abx@field@entrykey}{\blx@anchors}
138             {}
139             {\listxadd\blx@anchors{\the\c@refsection @\abx@field@entrykey}}}%
140     }
141 }
142 </package>
143 <*latex-lab>
144 \ProvidesFile{bib-latex-lab-testphase.ltx}
145     [\ltxlabbibdate\space v\ltxlabbibversion\space
146     latex-lab wrapper bib]
147
148 \RequirePackage{latex-lab-testphase-bib}
149
150 </latex-lab>

```