

Make link targets hyperref bundle

The L^AT_EX Project*

Version 2026-01-29 v7.01p

1 Commands to create and adapt targets

This module provides commands to create targets. Their goal is to unify and replace a number of user and internal hyperref commands and to provide a clear interface for package authors and users.

1.1 The main command

<code>\MakeLinkTarget</code>	<code>\MakeLinkTarget[<i><prefix></i>]{<i><counter></i>}</code>
	<code>\MakeLinkTarget[<i><prefix></i>]{}</code>
	<code>\MakeLinkTarget*{<i><manual target></i>}</code>

`\MakeLinkTarget` creates a target for an internal link, (called a destination if a PDF is created). In vertical mode the target is created where the command is issued, in horizontal mode it is typically (in a PDF) raised by the current `\normalbaselineskip` (similar to the targets created by `\refstepcounter`).

The arguments allow to control the name of the target/anchor/destination. While technically any unique name (e.g. some number) would work, names related to the actual counter both simplifies the debugging and referencing destinations from external documents. Also `\autoref` makes use of the name to identify the type of a label: it splits off the part until the first period and then looks up for a defined name. So if you create a target name `ABC.CDE.1.2`, `\autoref` tries to use `\ABCautorefname`.

The name is then stored *globally*¹ in `\@currentHref` and used for example in the next `\label`. A target name must be unique across a document. It is up to the users and package authors to ensure this by following the advice given below.

*E-mail: latex-team@latex-project.org

¹This means that the deprecated hyperref option `localanchorname` is ignored.

A global `\@currentHref` means that the target name and the label name can get out of sync: if for example a numbered equation is used between a section and a `\label`, a reference will show the section number but a link will jump to the equation.

`hyperref` allows to switch to local assignment of `\@currentHref` and so to align with the other `\label` values with the (so-called experimental) option `localanchorname`. But while on a document level this can work, it makes it difficult for package authors to add reliable link targets that can be referenced if it is unclear if the assignment is local or global.

So the choice was made to set `\@currentHref` globally always (and to deprecate `localanchorname`): This is the default in `hyperref` since ever and it didn't lead to major problems. It has the advantage to allow to put an anchor in a box and move it around. E.g. in the following example the two `\ref` commands reference the section but the second jumps to the top of the rule:

```
% \section{An example}\label{sec:title}
% \newpage \raisebox{3cm}[0pt][0pt]{\phantomsection}%
% \rule{1cm}{3cm}\label{sec:page2}
% page two
% \newpage
% \ref{sec:title}, \ref{sec:page2}
%
```

Target names from counters

`\MakeLinkTarget[⟨prefix⟩]{⟨counter⟩}`

If the mandatory argument is not empty it is interpreted as a `LATEX` counter name. The counter is not stepped by the command. If the `⟨counter⟩` doesn't exist, a warning is issued and no target is created.

The target name is created as expansion of `⟨prefix⟩.\theH⟨counter⟩` (and so both the prefix and `\theH⟨counter⟩` should be expandable). The default prefix is the counter name `⟨counter⟩`.

For example:

```
\MakeLinkTarget{section} ⇒ section.\theHsection ⇒ section.1.1
\MakeLinkTarget[sec]{section} ⇒ sec.\theHsection
```

`\theH⟨counter⟩` must be defined: `\MakeLinkTarget` does not try like other `hyperref` commands to construct the command on-the-fly but warns and creates no target if it doesn't exist. Be aware that if `hyperref` is used with the option `implicit=false` it does not predefine and create `\theH⟨counter⟩` representations so targets could be missing!²

Typically `\theH⟨counter⟩` should expand to numbers and periods and make use of parent counters to give a unique representation. The prefix can be a more or less arbitrary string. Spaces are allowed (but not really recommended). A star at the end should be avoided to prevent clashes with the names created when the internal counter is used. The use of non-ASCII chars with `pdflatex` depends on the `LATEX` version: With newer version many of them are safe and with a format 2022-06-01 or newer it should be even possible to use chars in a target name which are undeclared and can't be typeset in the document. But despite the fact that it works, it is recommended to stick to ASCII and to avoid spaces: It is unclear if all PDF viewers and editors can handle them, also they give rather unreadable names in the PDF like `(\360\237\246\206)` and so make debugging harder.

²It is planned that `LATEX` directly defines the `\theH⟨counter⟩` for all counters created with `\newcounter`.

Using a special prefix can be useful if the actual counter has an internal name like e.g. `tcb@cnt@example`. `\MakeLinkTarget[tcbexample]{tcb@cnt@example}` then gives a nicer looking target name. Care should be taken not to clash with existing counter names and names for `\autoref` should be adjusted if needed.

Targets using the internal counter

```
\MakeLinkTarget[prefix]{}
```

If the mandatory argument is empty a target name based on an internal absolute counter is created. The counter is stepped at every call (and also by other `hyperref` commands). The prefix is added with a star and a period (see above for the allowed chars). The default prefix is `page`.

```
\MakeLinkTarget{}           ⇒ page*.1
\MakeLinkTarget{}           ⇒ page*.2
\MakeLinkTarget[section]{} ⇒ section*.3
```

As an example the `hyperref` command `\phantomsection` is equivalent to `\MakeLinkTarget[section]`

Manual target names

```
\MakeLinkTarget*{manual target name}
```

If the starred variant is used a manual target name is created.

`\MakeLinkTarget*{destname}` is roughly equivalent to `\hypertarget{destname}` but unlike the latter it also raises the destination in a PDF, there is no text argument, and—most importantly—it updates `@currentHref`, that means the target name is stored by the next `\label`, something that `\hypertarget` doesn't do). The target name is expanded, see the remarks about prefixes above regarding the allowed chars.

When creating manually targets care should be taken to avoid clashes with automatic target names. As all automatic targets contain at least one period, names without a period are recommended.

The second (text) argument of `\hypertarget` is only used if the option `nesting` is set to true—but this option isn't used anywhere (in PDF the idea of a text in the anchor or nesting of anchors makes no sense anyway but also when html is produced e.g. with `make4ht` it is not used). So probably the option will be deprecated and removed.

1.2 Manipulate the next target name

Targets are sometimes created in places where it is difficult to inject a label to retrieve the target name for use e.g. in a bookmark. The next command allows to change the next target name:

```
\NextLinkTarget \NextLinkTarget*{manual target}
```

The command changes the next target name to `<manual target>`. It does the same as the `\hypersetup` key `next-anchor` and also affects targets created by `\refstepcounter`.

A use case are bookmarks for the table of contents where you know that the heading will create a target:

```
% \documentclass{book}
% \usepackage{bookmark}
%
% \begin{document}
% \bookmark[dest=toc]{Table of Contents}
% \NextLinkTarget{toc}
% \tableofcontents
% \chapter{A}
% \end{document}
%
```

Hooks

`\makelinktarget` The hook `\makelinktarget`³ is executed at the begin of the commands. It is inside a group and so can be used to locally change settings. See below for an example.

Suppressing the target

```
\LinkTargetOn \LinkTargetOn
\LinkTargetOff \LinkTargetOff
```

This commands allows to switch on and off locally the creation of a target with `\MakeLinkTarget`. The switches are also honored by `\refstepcounter`⁴. This allows to suppress the target from an internal `\refstepcounter` and replace it by some manual version by using grouping:

```
% \LinkTargetOff %suppress anchor in internal refstepcounter
% ...
% \refstepcounter{...}
% ...
% {\LinkTargetOn\MakeLinkTarget*{mytarget}} %create manual anchor
% ...
% \LinkTargetOn
%
```

Raising the target

In horizontal mode the target is raised by the current value of `\normalbaselineskip`.

To change this the hook can be used e.g. to double the value everywhere:

```
% \AddToHook{cmd/MakeLinkTarget/before}
%   {\setlength\normalbaselineskip{2\normalbaselineskip}}
% \leavevmode\MakeLinkTarget{section}
%
```

³The hook uses a plain name without reference to the `hyperref` package in anticipation of the move of this code into the L^AT_EX kernel.

⁴currently `\refstepcounter` doesn't use `\MakeLinkTarget` itself but this will probably change.

1.3 Changing all target names

`\SetLinkTargetFilter` `\SetLinkTargetFilter{{filter code using #1}}`

`hyperref` provides the command `\HyperDestNameFilter` to change all target names. It is applied⁵ to every target name and is also used in references, but it doesn't change `\@currentHref` itself. So after

```
% \renewcommand*{\HyperDestNameFilter}[1]{docA-#1}
%
```

you would get in the PDF everywhere the prefix `docA`

```
% %destination names:
% <<
% /Names [(docA-Doc-Start) 7 0 R (docA-chapter.1) 8 0 R (docA-page.1) 6 0 R]
% /Limits [(docA-Doc-Start) (docA-page.1)]
% >>
% %link to a chapter
% /A << /S /GoTo /D (docA-chapter.1) >>
% %link from the bookmark
% << /S /GoTo /D (docA-chapter.1) >>
%
```

but the label info in the `.aux` would show only `chapter.1`:

```
% \newlabel{chap}{1}{1}{Title}{chapter.1}{}
%
```

and so `\autoref` is still able to extract the counter name.

`\MakeLinkTarget` uses this filter too: it would break internal link commands if it would ignore it. To stay compatible with future development the filter should not be redefined directly but be set with `\SetLinkTargetFilter`. The command can only be used in the preamble.

```
% \SetLinkTargetFilter{docA-#1}
%
```

2 Implementation

```
1 <@@=hyp>
2 <*header>
3 \ProvidesExplPackage{hyperref-linktarget}{2022-06-20}{v7.00s}
4 {Making targets, destinations and anchors -- module of hyperref}
5 </header>
6 <package-include>\ExplSyntaxOn
```

⁵In the backend code, so it depends actually on the driver if it is honored or not

2.1 Variables

7 `<*package>`

`\l__hyp_target_create_bool` This boolean decides if a target is created at all. (it will replace `\@skiphyperref` long term)

```
8 \bool_new:N \l__hyp_target_create_bool
9 \bool_set_true:N \l__hyp_target_create_bool
```

(End of definition for `\l__hyp_target_create_bool`.)

`makelinktarget` This hook is used to adapt for example the raising

```
10 \hook_new:n {makelinktarget}
```

(End of definition for `makelinktarget`. This function is documented on page 4.)

2.2 Helper commands

`__hyp_target_raise:n` We need a command to raise the targets. It is mostly a copy from the `hyperref` command but we removed the hooks and use `\normalbaselineskip`. TODO: The code to save/restore the space factor should be replaced by kernel methods. change 2024-11-06: add a kern, see `thm-anchor` test.

```
11 \cs_new_protected:Npn \__hyp_target_raise:n #1
12 {
13   \mode_if_vertical:TF
14     { #1 }
15     {
16       \Hy@SaveSpaceFactor
17       \penalty\@M
18       \smash
19       {
20         \box_move_up:nn
21         { \normalbaselineskip }
22         {
23           \hbox:n
24           {
25             \Hy@RestoreSpaceFactor
26             #1
27             \Hy@SaveSpaceFactor
28           }
29         }
30       }
31       \kernOpt
32       \Hy@RestoreSpaceFactor
33     }
34 }
```

(End of definition for `__hyp_target_raise:n`.)

2.3 Providing the commands

In anticipation of the addition of the main commands to the kernel as no-ops we provide them:

```
35 \ProvideDocumentCommand\LinkTargetOn{}{}
36 \ProvideDocumentCommand\LinkTargetOff{}{}
37 \ProvideDocumentCommand\MakeLinkTarget{s0{}m}{}
38 \ProvideDocumentCommand\NextLinkTarget{m}{}
```

2.4 Target on and off switch

`\LinkTargetOn`
`\LinkTargetOff`

```
39 \RenewDocumentCommand\LinkTargetOn {}
40 {
41   \bool_set_true:N \l__hyp_target_create_bool
42 }
43
44
45 \RenewDocumentCommand\LinkTargetOff {}
46 {
47   \bool_set_false:N \l__hyp_target_create_bool
48 }
```

(End of definition for \LinkTargetOn and \LinkTargetOff. These functions are documented on page 4.)

`\MakeLinkTarget` This is the main command. To keep it simple we allow an optional argument also for the manual command but ignore it for now.

```
49 \RenewDocumentCommand\MakeLinkTarget {s 0{} m}
50 {
51   \bool_if:NT \l__hyp_target_create_bool
52   {
53     \group_begin:
54     \hook_use:n { makelinktarget }
55     \IfBooleanTF {#1}
56     {
57       \__hyp_target_manual:nn {#2}{#3}
58     }
59     {
60       \__hyp_target_counter:nn {#2}{#3}
61     }
62     \group_end:
63     \UseTaggingSocket{recordtarget}
64   }
65 }
```

(End of definition for \MakeLinkTarget. This function is documented on page 1.)

`__hyp_target_manual:nn` This is the code for the manual target name. The prefix is simply ignored.

```
66 \cs_new_protected:Npn \__hyp_target_manual:nn #1 #2 %#1 prefix, #2 name
67 {
68   \tl_gset:Ne \@currentHref {#2}
69   \hook_use:n {\__hyp/target/setname}
```

```

70   \@onelevel@sanitize\@currentHref
71   \__hyp_target_raise:n {\hyper@anchorstart{\@currentHref}\hyper@anchorend}
72 }

```

(End of definition for __hyp_target_manual:nn.)

__hyp_target_counter:nn The code for counter related targets must be split into the case where the internal counter is used, and where a user counter is used

```

73 \cs_new_protected:Npn \__hyp_target_counter:nn #1 #2  % #1 prefix, #2 counter or empty
74 {
75   \tl_if_blank:nTF {#2}
76   {
77     \__hyp_target_counter_anon:n {#1}
78   }
79   {
80     \__hyp_target_counter_doc:nn {#1}{#2}
81   }
82 }
83

```

(End of definition for __hyp_target_counter:nn.)

__hyp_target_counter_anon:n This creates the target with the internal count. We use the same (tex) count `\Hy@linkcounter` than the other hyperref commands.

```

84 \cs_new_protected:Npn \__hyp_target_counter_anon:n #1
85 {
86   \int_gincr:N\Hy@linkcounter
87   \tl_gset:Ne \@currentHref
88   {\tl_if_blank:nTF{#1}{page}{#1}*.\int_use:N\Hy@linkcounter}
89   \hook_use:n {__hyp/target/setname}
90   \@onelevel@sanitize\@currentHref
91   \__hyp_target_raise:n {\hyper@anchorstart{\@currentHref}\hyper@anchorend}
92 }

```

(End of definition for __hyp_target_counter_anon:n.)

__hyp_target_counter_doc:nn And now the target with the user counter. We warn if the counter or the representation doesn't exist

```

93 \cs_new_protected:Npn \__hyp_target_counter_doc:nn #1 #2
94 {
95   \bool_lazy_or:nnTF { \cs_if_free_p:c {c@#2} } { \cs_if_free_p:c {theH#2} }
96   {
97     \PackageWarning {hyperref}{Counter~'#2'~or~the~representation~'\string\theH#2'\Message
98     don't~exist.~No~target~created.}{}
99   }
100  {
101    \tl_gset:Ne \@currentHref {\tl_if_blank:nTF{#1}{#2}{#1}.\use:c{theH#2}}
102    \hook_use:n {__hyp/target/setname}
103    \@onelevel@sanitize\@currentHref
104    \__hyp_target_raise:n {\hyper@anchorstart{\@currentHref}\hyper@anchorend}
105  }
106 }

```


(End of definition for _hyp_target_counter_doc:nn.)

\NextLinkTarget we rely on the internal hook to set the next target name:

```
107 \RenewDocumentCommand\NextLinkTarget {m}
108   {
109     \hook_gput_next_code:nn {__hyp/target/setname}
110     {
111       \tl_gset:Nc \@currentHref {#1}
112     }
113   }
114
```

(End of definition for \NextLinkTarget. This function is documented on page 3.)

\SetLinkTargetFilter This is an interface to \HyperDestNameFilter

```
115 \NewDocumentCommand\SetLinkTargetFilter {m}
116   {
117     \cs_set:Npn \HyperDestNameFilter ##1 {#1}
118   }
119 \@onlypreamble \SetLinkTargetFilter
```

(End of definition for \SetLinkTargetFilter. This function is documented on page 5.)

```
120 \endpackage
121 \end{@@=}.
122 \package-include\ExplSyntaxOff
```