# The latex-lab-marginpar package
# Changes related to the tagging of margin notes

LaTeX Project*

v0.85j 2026-08-12

**Abstract**

## 1 Introduction

This module contains changes to improve the tagging (in the standard classes) of margin notes created with the `\marginpar` command.

Such margin notes are rather small but nevertheless tagging is not trivial and poses a number of interesting problems both regarding the structure and the implementation.

### 1.1 Structure

`\marginpar` creates small boxes in the margin of a page. While they are technically floats they also typically relate to the paragraph beside them, so their structure element should be placed near such a paragraph.

They can be tagged either as artifacts (if they are merely distracting decoration), as small headings before the paragraph, or as `Aside`. Unlike the PDF 1.7 fallback `Note` the structure `Aside` is not allowed inside `P`, so if `Aside` is used, it must be placed before or after the current `P` in the surrounding `semantic-para`, or it must split the `P`. Splitting is probably not so good as `\marginpar` is often used somewhere in the middle of sentence. The best default is probably to use `Aside`, find the parent `semantic-para` and add it there.

### 1.2 Implementation

`marginpar` has an optional argument which allows to define a different content on left/right margins for odd/even pages (depending on twoside and `\reversemarginpar`).

The current implementation stores *both* arguments in boxes and decides in the output routine which one to use. This is quite problematic for tagging, as the unused box produces structure objects, literals, labels, MCID numbers, and perhaps OBJR-objects from links which must be later thrown away again (or are thrown away by the engine). While it is theoretically not impossible to create both boxes while taging is active but to keep everything on hold and insert the real structure only when the box is used it is in practice quite difficult, slow and error prone.

There are a number of options to avoid this hassle.

*Initial implementation done by Ulrike Fischer

**Minimal tagging** One option is to disable tagging when the boxes are built and when the box is used to surround them with a simple Aside.

**Pro:** easy to implement

**Contra:** Content that perhaps needed tagging (`\LaTeX` logo with /ActualText, images, links) is not tagged.

**Remaking the box** Tagging is deactivated when the boxes are created. The arguments are stored and when the box should be used in the output routine (and when it is known which one will be used) the box is (re)created, now with proper tagging.

Contra: The content can change e.g. if it uses counters or macros that are redefine later before the output routine is called. As tagging is done later, it is not trivial to insert the `Aside` in the right place in the paragraph structure.

**Two pass compilation** A label is used to detect which argument/boxed is used and tagging is activated only for this one.

Contra: costs a `label` and requires a two pass compilation (which is needed anyway). This should be ok. `\marginnote` uses a label too, and the same label could also be used to resolve problems if a margin note moves to a new page (and so replace the mparhack package).

The last option seems the best and is therefore implemented.

# 2 Implementation

```
⟨*package⟩
⟨@@=tag⟩

\ProvidesExplPackage {latex-lab-testphase-marginpar} {\ltlabmarginpardate} {\ltlabmarginparver
  {Changes related to the tagging of the margin notes}
```

## 2.1 Variables

We need a variable to make the label unique. Todo: Not sure about the name.

\g__kernel_marginpar_int

```
\int_new:N \g__kernel_marginpar_int
```

(*End of definition for* `\g__kernel_marginpar_int`.)

## 2.2 Tagging sockets

upport/marginpar/begin (*socket*)
gsupport/marginpar/end (*socket*)

The sockets are declared in lttagging.

```
%\socket_new:nn {tagsupport/marginpar/begin}{0}
%\socket_new:nn {tagsupport/marginpar/end}{0}
```

__tag_get_marginpar_parent:N
__tag_get_marginpar_parent_aux:nn

In a paragraph we must retrieve the structure number of the surrounding Part structure TODO: if this is needed in more places this should go into tagpdf.

```
\cs_new_protected:Npn\__tag_get_marginpar_parent_aux:nn #1 #2
 {
   \str_if_eq:eeT{\use_ii:nn #1}{Part}{\seq_put_right:Nn\l__tag_tmpa_seq {#2}}
```

```
    \str_if_eq:eeT{\use_ii:nn #1}{Sect}{\seq_put_right:Nn\l__tag_tmpa_seq {#2}}
  }
\cs_new_protected:Npn \__tag_get_marginpar_parent:N #1
  {
    \seq_clear:N\l__tag_tmpa_seq
    \tag_struct_map_function:N \__tag_get_marginpar_parent_aux:nn
    \seq_put_right:Nn\l__tag_tmpa_seq {2}
    \seq_get_left:NN \l__tag_tmpa_seq #1
  }
\cs_generate_variant:Nn \tag_struct_begin:n {e}
```

(*End of definition for* `\__tag_get_marginpar_parent:N` *and* `\__tag_get_marginpar_parent_aux:nn`.)

gsupport/marginpar/begin (*plug*)
tagsupport/marginpar/end (*plug*)

```
\NewTaggingSocketPlug{marginpar/begin}{default}
  {
   \if_mode_horizontal:
    \tag_mc_end:
    \__tag_get_marginpar_parent:N \l__tag_tmpa_tl
    \tag_struct_begin:e{tag=\UseStructureName{marginnote},parent=\l__tag_tmpa_tl}%
   \else:
    \tag_struct_begin:n{tag=\UseStructureName{marginnote}}%
   \fi:
  }
\NewTaggingSocketPlug{marginpar/end}{default}
  {
    \tag_struct_end:
    \if_mode_horizontal: \tag_mc_begin:n{} \fi:
  }
\AssignTaggingSocketPlug{marginpar/begin}{default}
\AssignTaggingSocketPlug{marginpar/end}{default}
```

## 2.3 Kernel command changes

\@savemarbox

We add sockets that add a tagging structure Aside:

```
\long\def \@savemarbox #1#2{%
  \UseTaggingSocket{marginpar/begin}
  \global\setbox #1%
    \color@vbox
      \vtop{%
        \hsize\marginparwidth
        \@parboxrestore
        \@marginparreset
        #2\par
        \@minipagefalse
        \outer@nobreak
        }%
    \color@endbox
   \UseTaggingSocket{marginpar/end}
}
```

(*End of definition for* `\@savemarbox`.)

\@ympar

We must avoid that `\@xympar` creates tagging structure:

```
\long\def\@ympar#1{%
```

```
  \@savemarbox\@marbox{#1}%
  \global\setbox\@currbox\copy\@marbox
  \tag_suspend:n{\@ympar}
  \@xympar
  \tag_resume:n{\@ympar}}
```

(*End of definition for* `\@ympar`.)

`\@xmpar` `\@xmpar` is the command used if an optional argument is present. In this case we must stop tagging for `\@xympar` as above but also decide which of the two @savemarbox should have tagging active. We add a label to `\@marbox`. If that exist tagging is activated for this box else for the other one.

```
\long\def\@xmpar[#1]#2{%
  \int_gincr:N\g__kernel_marginpar_int
  \property_if_recorded:eTF { tag_marginpar-opt-\int_use:N\g__kernel_marginpar_int }
    {
     \@savemarbox\@marbox {#1
       \property_record:ee
         { tag_marginpar-opt-\int_use:N\g__kernel_marginpar_int }{page}
       }%
     \tag_suspend:n{\@xmpar}
     \@savemarbox\@currbox{#2}%
     \tag_resume:n{\@xmpar}
    }
```

order matters! the tagged box should be first so that it can pick up the correct semantic-para number.

```
    {
     \@savemarbox\@currbox{#2}%
     \tag_suspend:n{\@xmpar}
     \@savemarbox\@marbox{#1
       \property_record:ee
         { tag_marginpar-opt-\int_use:N\g__kernel_marginpar_int }{page}}
     \tag_resume:n{\@xmpar}
    }
  \tag_suspend:n{\@xmpar}
  \@xympar
  \tag_resume:n{\@xmpar}
  }
⟨/package⟩
```

(*End of definition for* `\@xmpar`.)