

# The latex-lab-amsmath code\*

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

May 19, 2025

Abstract

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>The Implementation</b>	<b>1</b>
2.1	File declaration . . . . .	2
2.2	Tagpdf support . . . . .	2
2.3	Measuring . . . . .	2
2.4	Display environments . . . . .	2
2.4.1	Tag . . . . .	2
2.4.2	align & friends . . . . .	3
2.4.3	gather and gather* . . . . .	5
2.4.4	gathered, aligned and alignedat . . . . .	7
2.4.5	multiline and multiline* . . . . .	8
2.5	Cases . . . . .	11
2.5.1	smallmatrix . . . . .	11
2.6	The split environment . . . . .	12
2.7	\intertext . . . . .	17
2.8	\text . . . . .	18
2.9	\pmb . . . . .	19

<b>Index</b>	<b>20</b>
--------------	-----------

## 1 Introduction

This file implements adaptations to the `amsmath` package needed for the tagging project.

---

\*

## 2 The Implementation

Better no @@ expansion here . . . . .

```
1 <@@= >
2 <*kernel >
```

### 2.1 File declaration

```
3 \ProvidesFile{latex-lab-amsmath.ltx}
4 [2025-03-07 v0.1f amsmath adaptions]
```

### 2.2 Tagpdf support

To make the code independent from tagging being loaded and active we load the tagpdf-base package:

```
5 \RequirePackage{tagpdf-base}
6 \ExplSyntaxOn
```

### 2.3 Measuring

When measuring we neither want tagging nor the luamml processing.

```
\measuring@true
7 \def\measuring@true{\let\ifmeasuring@\iftrue\tag_suspend:n{\measuring}\luamml_ignore:}
(End of definition for \measuring@true. This function is documented on page ??.)
```

### 2.4 Display environments

#### 2.4.1 Tag

The tag/label must be saved, so that it can be reinserted later.

TODO: \maketag@@@ is perhaps used in places where tagging/luamml handling is not wanted. This must be checked and handled.

```
\maketag@@@
8 \def\maketag@@@#1
9   {%
10     \ifmeasuring@
11       \hbox{\m@th\normalfont#1}%
12     \else
13       \UseTaggingSocket{math/display/tag/begin}
14       \hbox{\m@th\normalfont#1
15         \UseTaggingSocket{math/luamml/mtable/tag/save}
16       }%
17       \UseTaggingSocket{math/display/tag/end}
18     \fi
19   }
```

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

\eqref uses \tagform@ and so \maketag@@@ but we do not want this tagging there.

```

\eqref
\maketag@@@notog 20 \def\maketag@@@notag#1{\hbox{\m@th\normalfont#1}}
21 \DeclareRobustCommand{\eqref}[1]
22   {\textup{\let\maketag@@@\maketag@@@notag\tagform@{\ref{#1}}}}

```

(End of definition for `\eqref` and `\maketag@@@notog`. These functions are documented on page ??.)

## 2.4.2 align & friends

Most display environment uses a common command for the end which contains the `luamml` socket to finalize the mtable.

```

\common@align@ending
23 \def\common@align@ending {
24   \math@cr \black@totwidth@
25   \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {\@currenvir}
26   \egroup
27   \ifingather@
28     \restorealignstate@
29     \egroup
30     \nonumber
31     \ifnum0='{ \fi \iffalse } \fi
32   \else
33     \dollar@end
34     \fi
35     \ignorespacesafterend
36 }

```

(End of definition for `\common@align@ending`. This function is documented on page ??.)

Now we redefine the display alignments to use these ending.

```

37 \renewenvironment{align}{%
38   \start@align\@ne\st@rredfalse\m@ne
39 }{%
40   \common@align@ending
41 }
42 \renewenvironment{align*}{%
43   \start@align\@ne\st@rredtrue\m@ne
44 }{%
45   \common@align@ending
46 }
47 \renewenvironment{alignat}{%
48   \start@align\z@\st@rredfalse
49 }{%
50   \common@align@ending
51 }
52 \renewenvironment{alignat*}{%
53   \start@align\z@\st@rredtrue
54 }{%
55   \common@align@ending
56 }
57 \renewenvironment{flalign}{%
58   \start@align\tw@\st@rredfalse\m@ne
59 }{%
60   \common@align@ending

```

```

61 }
62 \renewenvironment{flalign*}{%
63   \start@align\tw@\st@rredtrue\m@ne
64 }{%
65   \common@align@ending
66 }
67 \renewenvironment{xalignat}{%
68   \start@align\@ne\st@rredfalse
69 }{%
70   \common@align@ending
71 }
72 \renewenvironment{xalignat*}{%
73   \start@align\@ne\st@rredtrue
74 }{%
75   \common@align@ending
76 }
77 \renewenvironment{xxalignat}{%
78   \start@align\tw@\st@rredtrue
79 }{%
80   \common@align@ending
81 }

```

And register these environments for the math grabbing.

```

82 \math_register_halign_env:n {align}{}
83 \math_register_halign_env:n {align*}{}
84 \math_register_halign_env:n {alignat}{}
85 \math_register_halign_env:n {alignat*}{}
86 \math_register_halign_env:n {flalign}{}
87 \math_register_halign_env:n {flalign*}{}
88 \math_register_halign_env:n {xalignat}{}
89 \math_register_halign_env:n {xalignat*}{}
90 \math_register_halign_env:n {xxalignat}{}

```

The align preamble (used in \align@) needs code for luamml to save the cells.

\align@preamble

```

91 \def\align@preamble{%
92   &\hfil
93   \strut@
94   \setboxz@h
95   {
96     \@lign
97     $
98     \m@th\displaystyle{##}
99     \ifmeasuring@
100     \luamml_ignore:
101     \else
102     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
103     \fi
104     $
105   }%
106 \ifmeasuring@
107   \savefieldlength@
108 \else
109   \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}

```

```

110 \fi
111 \set@field
112 \tabskip\z@skip
113 &\setboxz@h
114 {
115 \@lign
116 $
117 \m@th\displaystyle{##}
118 \ifmeasuring@
119 \luamml_ignore:
120 \else
121 \UseTaggingSocket{math/luamml/save/nNn}{ } \displaystyle {mtd} }
122 \fi
123 $
124 }%
125 \ifmeasuring@
126 \savefieldlength@
127 \else
128 \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
129 \fi
130 \set@field
131 \hfil
132 \tabskip\alignsep@
133 }

```

(End of definition for `\align@preamble`. This function is documented on page ??.)

`\math@cr@@@align`

```

134 \def\math@cr@@@align{%
135 \ifst@rred\nonumber\fi
136 \if@eqnsw \global\tag@true \fi
137 \global\advance\row@ \@one
138 \add@amps\maxfields@
139 \omit
140 \kern-\alignsep@
141 \iftag@
142 \setboxz@h{\@lign\strut@{\make@display@tag}}%
143 \place@tag
144 \fi
145 \UseTaggingSocket{math/luamml/mtable/tag/set}
146 \ifst@rred\else\global\@eqnswtrue\fi
147 \global\lineht@\z@
148 \cr
149 }

```

(End of definition for `\math@cr@@@align`. This function is documented on page ??.)

### 2.4.3 gather and gather\*

`\gather@`

```

150 \def\gather@#1{%
151 \ingather@true \let\split\insplit@
152 \let\tag\tag@in@align \let\label\label@in@display
153 \chardef\dspbrk@context\z@
154 \intertext@ \displ@y@ \Let@

```

```

155 \let\math@cr@@\math@cr@@gather
156 \gmeasure@{#1}%
157 \global\shifttag@false
158 \tabskip\z@skip
159 \global\row@\@ne
160 \halign to\displaywidth\bgroup
161   \strut@
162   \setboxz@h
163   {
164     $\m@th\displaystyle{##}
165     \UseTaggingSocket{math/luamml/save/nNn}{ { } \displaystyle {mtd} }
166     $
167   }%
168   \UseTaggingSocket{math/luamml/mtable/finalizecol}{box}
169   \calc@shift@gather
170   \set@gather@field
171   \tabskip\@centering
172   &\setboxz@h{\strut@{##}}%
173   \place@tag@gather
174   \UseTaggingSocket{math/luamml/mtable/tag/set}
175   \tabskip \iftagsleft@ \gdisplaywidth@ \else \z@skip \span\fi
176   \crr
177   #1%
178 }

```

(End of definition for `\gather@`. This function is documented on page ??.)

`\endgather`

```

179 \def\endgather{
180   \math@cr
181   \black@ \totwidth@
182   \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
183   \egroup
184   \dollar@end
185   \ignorespacesafterend
186 }

```

(End of definition for `\endgather`. This function is documented on page ??.)

The original definition of `gather*` uses `\endgather` but this redirection doesn't work if we alter `gather` so we use the real meaning and add the socket.

`gather*` (*env.*)

```

187 \renewenvironment{gather*}
188 {
189   \start@gather\st@rredtrue
190 }
191 {
192   \math@cr
193   \black@\totwidth@
194   \UseExpandableTaggingSocket{math/luamml/mtable/finalize} {gather}
195   \egroup
196   \dollar@end
197   \ignorespacesafterend
198 }

```

Register both environments

```
199 \math_register_halign_env:nn {gather}{}  
200 \math_register_halign_env:nn {gather*}{}
```

#### 2.4.4 gathered, aligned and alignedat

These environments are not grabbed as they are inside other display environments but they need various sockets for luamml support.

`\start@aligned`

```
201 \renewcommand{\start@aligned}[2]{  
202   \RIfM@  
203   \else  
204     \nonmatherr@ {\begin{\@currenvir}}  
205   \fi  
206   \savecolumn@ % Assumption: called inside a group  
207   \UseTaggingSocket{math/luamml/annotate/false}{}{ \alignedspace@left }  
208   \ams@start@box {#1} \bgroup  
209     \maxfields@ #2 \relax  
210     \ifnum \maxfields@ > \m@ne  
211       \multiply \maxfields@ \tw@  
212       \let \math@cr@@@ \math@cr@@@alignedat  
213       \alignsep@ \z@skip  
214     \else  
215       \let \math@cr@@@ \math@cr@@@aligned  
216       \alignsep@ \minalignsep  
217     \fi  
218     \Let@ \chardef \dspbrk@context \@ne  
219     \default@tag  
220     \spread@equation % no-op if already called  
221     \global \column@ \z@  
222     \ialign \bgroup  
223       & \column@plus  
224       \hfil  
225       \strut@  
226       $  
227         \m@th  
228         \displaystyle  
229         {##}  
230         \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }  
231       $  
232       \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}  
233       \tabskip \z@skip  
234     & \column@plus  
235     $  
236       \m@th  
237       \displaystyle  
238       {  
239         {}  
240         ##  
241       }  
242     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }  
243     $  
244     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
```

```

245     \hfil
246     \tabskip\alignsep@
247     \crcr
248     \ams@return@opt@arg
249 }

```

(End of definition for `\start@aligned`. This function is documented on page ??.)

`gathered` (*env.*)

```

250 \renewenvironment{gathered}[1][c]{%
251   \RIfM@else
252   \nonmatherr@{\begin{gathered}}%
253   \fi

```

This annotates the space

```

254   \UseTaggingSocket{math/luamml/annotate/false } {}{ \alignedspace@left }
255   \ams@start@box{#1}\bgroup
256   \Let@ \chardef\dspbrk@context\@ne \restore@math@cr
257   \spread@equation
258   \ialign\bgroup
259     \hfil\strut@$\m@th\displaystyle##

```

This save the cell and then finalize it.

```

260     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
261     $
262     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
263     \hfil
264     \crcr
265     \ams@return@opt@arg
266 }{%
267   \endaligned
268 }

```

`\endaligned`

```

269 \def\endaligned
270 {
271   \crcr
272   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}
273   \egroup
274   \restorecolumn@
275   \egroup
276   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
277 }

```

(End of definition for `\endaligned`. This function is documented on page ??.)

#### 2.4.5 `multiline` and `multiline*`

`multiline*` needs a redefinition before it is registered as it uses `\endmultiline`, this must be replaced by the true code.

`multiline*` (*env.*)

```

278 \renewenvironment{multiline*}{\start@multiline\st@rredtrue}
279 {%
280   \iftagsleft@ \@xp\lendmultiline@ \else \@xp\rendmultiline@ \fi

```



```

281 \ignorespacesafterend
282 }

```

And now we register both

```

283 \math_register_halign_env:nn {multiline}{}
284 \math_register_halign_env:nn {multiline*}{}

```

In the internal commands we have to add sockets for alignment attributes

`\multiline@`

```

285 \def\multiline@#1{%
286   \Let@
287   \@display@init{\global\advance\row@\@ne \global\dspbrk@lvl\m@ne}%
288   \chardef\dspbrk@context\z@
289   \restore@math@cr
290   \let\tag\tag@in@align
291   \global\tag@false \global\let\raise@tag\@empty
292   \mmeasure@{#1}%
293   \let\tag@gobble@tag \let\label\@gobble
294   \tabskip \if@fleqn \@mathmargin \else \z@skip \fi
295   \totwidth@\displaywidth
296   \if@fleqn
297     \advance\totwidth@-\@mathmargin
298   \fi
299   \halign\bgroup
300     \hbox to\totwidth@{%
301       \if@fleqn
302         \hskip \@centering \relax
303       \else
304         \hfil
305       \fi
306       \strut@
307       $\m@th\displaystyle{##}\endmultiline@math
308       \hfil
309     }%
310     \crrc
311     \if@fleqn
312       \hskip-\@mathmargin
313     \def\multiline@indent{\hskip\@mathmargin}% put it back
314   \else
315     \hfilneg
316     \def\multiline@indent{\hskip\multilinegap}%
317   \fi
318   \iftagsleft@
319     \iftag@
320       \begingroup
321         \ifshifftag@
322           \rlap{\vbox{%
323             \normalbaselines
324             \hbox{%
325               \strut@
326               \make@display@tag
327             }%
328             \vbox to\lineht@{%
329               \raise@tag

```

```

330         }}%
331         \multline@indent
332     \else
333         \setbox\z@\hbox{\make@display@tag}%
334         \dimen@\@mathmargin \advance\dimen@-\wd\z@
335         \ifdim\dimen@<\multlinetaggap
336             \dimen@\multlinetaggap
337         \fi
338         \box\z@ \hskip\dimen@\relax
339     \fi
340 \endgroup
341 \else
342     \multline@indent
343 \fi
344 \else
345     \multline@indent
346 \fi
347 \ifmeasuring@ \else
348     \UseTaggingSocket{math/luamml/mtable/aligncol} {left}
349 \fi
350 #1%
351 \ifmeasuring@ \else
352     \UseTaggingSocket{math/luamml/mtable/aligncol} {right}
353 \fi
354 }

```

(End of definition for `\multline@`. This function is documented on page ??.)

Luckily, `\multline` uses `\endmultline@math` in exactly the spot where we have to set the flag. Less luckily, `\endmultline@math` sometimes get overwritten for the last line. But that isn't much of a problem since we want special behavior there anyway.

`\endmultline@math`

```

355 \def\endmultline@math
356 {
357     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
358     $
359     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
360 }

```

(End of definition for `\endmultline@math`. This function is documented on page ??.)

`\rendmultline@`

```

361 \def\rendmultline@{%
362     \iftag@

```

we need to surround the math token with tagging sockets.

```

363     \UseTaggingSocket{math/luamml/save/nNn}{ {} \displaystyle {mtd} }
364     $
365     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
366     \let\endmultline@math\relax
367     \ifshifttag@
368         \hskip\multlinegap
369         \llap{\vtop{%
370             \raise@tag
371             \normalbaselines

```

```

372         \setbox\@ne\null
373         \dp\@ne\lineht@
374         \box\@ne
375         \hbox{\strut\make@display@tag}%
376     }}%
377     \else
378         \hskip\multlinetaggap
379         \make@display@tag
380     \fi

```

Here we set the tag TODO: is that sensible with multiline? Where is the tag saved?

```

381     \UseTaggingSocket{math/luamml/mtable/tag/set}
382     \else
383         \hskip\multlinegap
384     \fi
385     \hfilneg
386     \math@cr

```

Now we finalize the mtable.

```

387     \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
388     \egroup
389     \dollar@dollar@end
390 }

```

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

And something similar for the left version

`\lendmultiline@`

```

391 \def\lendmultiline@{%
392     \hfilneg
393     \hskip\multlinegap
394     \math@cr
395     \UseExpandableTaggingSocket {math/luamml/mtable/finalize} {multiline}
396     \egroup
397     \dollar@dollar@end
398 }

```

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

## 2.5 Cases

`env@cases`

```

399 \def\env@cases{%
400     \let\@ifnextchar\new@ifnextchar
401     \left\lbrace
402     \def\arraystretch{1.2}%
403     \array{ @{ } l @ { \quad } l @ { \luamml_ignore: } }%
404 }

```

*(End of definition for env@cases. This function is documented on page ??.)*

## 2.5.1 smallmatrix

`smallmatrix` (*env.*)

```

405 \renewenvironment {smallmatrix} {
406   \UseTaggingSocket{ math/luamml/annotate/false } {} { \null\, }
407   \vcenter \bgroup
408     \Let@
409     \restore@math@cr
410     \default@tag
411     \baselineskip 6 \ex@
412     \lineskip 1.5 \ex@
413     \lineskiplimit \lineskip
414     \ialign \bgroup
415       \hfil
416       $
417       \m@th
418       \scriptstyle
419       ##
420       % No \scriptsize here since we want to add the mstyle nodes
421       \UseTaggingSocket{math/luamml/save/nn}{ } {mtd} }
422       $
423       \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
424       \hfil
425     &&
426     \thickspace
427     \hfil
428     $
429     \m@th
430     \scriptstyle
431     ##
432     % No \scriptsize here since we want to add the mstyle nodes
433     \UseTaggingSocket{math/luamml/save/nn}{ } {mtd} }
434     $
435     \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}
436     \hfil
437   \crcr
438 }{ %
439   \crcr
440   \UseExpandableTaggingSocket{math/luamml/mtable/smallmatrix/save}
441   \egroup
442   \egroup
443   \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
444   \UseTaggingSocket{ math/luamml/annotate/false } {} { \, }
445 }
```

## 2.6 The split environment

The `split` environment is not trivial to handle as it has a rather convoluted implementation in `amsmath`: depending on in which display environment it is embedded it takes different branches, which makes it difficult to finalize the `mtable`.

The following patches work with `leqno` and `reqno` if the (default) `centertags` are used. The currently fail (the structure is wrong) if the option `tbtags` is used. The

alignment of the cells is currently not handled. A simple debug command until everything is sorted.

`\_math_split_debug_typeout:n`

446 `\cs_new:Npn\_math_split_debug_typeout:n#1{\use_none:n{#1}}`

(End of definition for `\_math_split_debug_typeout:n`.)

We need to detect if `\gather@split` has been used or not

`\l__math_gathersplit_bool`

447 `\bool_new:N\l__math_gathersplit_bool`

(End of definition for `\l__math_gathersplit_bool`.)

At first a redefinition of the main environment. Here we only have to add the saving command for the inner table:

`split (env.)`

```
448 \renewenvironment{split}{%
449   \_math_split_debug_typeout:n {begin~split}
450   \if@display
451     \ifinner
452       \@xp\@xp\@xp\split@aligned
453     \else
454       \ifst@rred \else \global\@eqnswtrue \fi
455     \fi
456   \else
457     \let\endsplit\@empty \@xp\collect@body\@xp\split@err
458   \fi
459   \collect@body\gather@split
460 }{\_math_split_debug_typeout:n{end~split}}%
461   \crrc
462   \UseExpandableTaggingSocket{math/luamml/mtable/innertable/save}%
463   \egroup
464 \egroup
465 \iftagsleft@ \@xp\lendsplit@ \else \@xp\rendsplit@ \fi
466 }
```

In `\gather@split` we have to add the finalization socket. We also set the boolean to true so that we can detect if the finalization has already happened. Perhaps this could be done in the luamml code instead?

`\gather@split`

```
467 \def\gather@split#1#2#3{
468   \_math_split_debug_typeout:n{gather@split}%
469   \@xp\endgroup \reset@equation % math@cr will handle equation numbering
470   \iftag@
471     \toks@\@xp{\df@tag}%
472     \edef\split@tag{%
473       \gdef\@nx\df@tag{\the\toks@}%
474       \global\@nx\tag@true \@nx\nonumber
475     }%
476   \else \let\split@tag\@empty
477   \fi
478   \bool_set_true:N\l__math_gathersplit_bool
479   \spread@equation
```

```

480 \vcenter\bgroup
481 \gather@{\split@tag
482 \begin{split}#1\end{split}}%
483 \def\endmathdisplay@a{%
484 \_math_split_debug_typeout:n{endmathdisplay@a}
485 \_math_split_debug_typeout:n{finalize~innertable~endmathdisplay@a}
486 \math@cr
487 \black@ \totwidth@
488 \egroup
489 \egroup
490 \UseExpandableTaggingSocket{math/luamml/mtable/innertable/finalize}%
491 }%
492 }

```

(End of definition for `\gather@split`. This function is documented on page ??.)

`\insplit@` In `\insplit@` we have to add the sockets which store the cell content.

```

493 \def\insplit@{\_math_split_debug_typeout:n{insplit@}}%
494 \global\setbox\z@\vbox\bgroup
495 \Let@ \chardef\dspbrk@context\@ne \restore@math@cr
496 \default@tag % disallow use of \tag here
497 \ialign\bgroup
498 \hfil
499 \strut@
500 $\m@th\displaystyle {##}%
501 \UseTaggingSocket{math/luamml/save/nNn}{ } \displaystyle {mtd} }%
502 $%
503 \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
504 &$\m@th\displaystyle { }##}%
505 \UseTaggingSocket{math/luamml/save/nNn}{ } \displaystyle {mtd} }%
506 $%
507 \UseTaggingSocket{math/luamml/mtable/finalizecol}{last}%
508 \hfill % Why not \hfil?---dmj, 1994/12/28
509 \crcr
510 }

```

(End of definition for `\insplit@`. This function is documented on page ??.)

And now the difficult part. Depending on the options `leqno/reqno` `\lendsplit@` or `\rendsplit@` are used for the typesetting and the inner table must be finalized here in case this hasn't happen yet. This must be tested with the boolean from `\gather@split`

```

511 \def\lendsplit@{%
512 \global\setbox9\vtop{\unvcopy\z@}%
513 \ifinalign@
514 \setbox\@ne\vbox{%
515 \unvcopy\z@
516 \global\setbox8\lastbox
517 }%
518 \setbox\@ne\hbox{%
519 \unhcopy8%
520 \unskip
521 \setbox\tw@\lastbox
522 \unskip
523 \global\setbox\thr@@\lastbox
524 }%

```

```

525     \_math_split_debug_typeout:n{lendsplit@/aligncase}
526     \ifctagsplit@
527     \_math_split_debug_typeout:n{lendsplit@/aligncase/centertags}
528     \gdef\split@{%
529         \hbox to\wd\thr@@{%
530             &\vcenter{\vbox{\moveleft\wd\thr@@\box9}}}%
531             \_math_split_debug_typeout:n{finalize~innertable~aligncase}
532             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
533         }%
534     \else
535     %TODO, not correct yet.
536     \_math_split_debug_typeout:n{lendsplit@/aligncase/tbtags}
537     \gdef\split@{%
538         \hbox to\wd\thr@@{%
539             &\vbox{\moveleft\wd\thr@@\box9}%
540             \_math_split_debug_typeout:n{finalize~innertable~aligncase}
541             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
542         }%
543     \fi
544     \else

```

If not in align we need to test for \gather@split

```

545     \ifctagsplit@
546     \bool_if:NTF \l__math_gathersplit_bool
547     {
548         \_math_split_debug_typeout:n{lendsplit/equationcase/centertags}
549         \gdef\split@%
550         {\UseTaggingSocket{math/luamml/annotate/false}{}\vcenter{\box9}}
551     }
552     {
553         \_math_split_debug_typeout:n {lendsplit/gathercase/centertags}
554         \gdef\split@{\vcenter{\box9}}%
555         \_math_split_debug_typeout:n {finalize~innertable~gathercase}
556         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
557     }
558     \else
559     % TODO tbtags not correct yet.
560     \bool_if:NTF \l__math_gathersplit_bool
561     {
562         \_math_split_debug_typeout:n {lendsplit/equationcase/tbtags}
563         \gdef\split@%
564         {\UseTaggingSocket{math/luamml/annotate/false}{}\box9}}
565     }
566     {
567         \_math_split_debug_typeout:n {lendsplit/gathercase/tbtags}
568         \gdef\split@{
569             \box9%
570             \_math_split_debug_typeout:n {finalize~innertable~gathercase}
571             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
572     }
573     \fi
574     \fi
575     \aftergroup\split@
576 }

```

`\rendsplit@` And more or less the same for the `\rendsplit@` environment.

```

577 \def\rendsplit@{%
578   \ifalign@
579     \global\setbox9 \vtop{%
580       \unvcopy\z@
581       \global\setbox8 \lastbox
582       \unskip
583     }%
584     \setbox\@ne\hbox{%
585       \unhcopy8
586       \unskip
587       \global\setbox\tw@\lastbox
588       \unskip
589       \global\setbox\thr@\@lastbox
590     }%
591     \ifctagsplit@
592       \gdef\split@{%
593         \hbox to\wd\thr@{%
594           &\vcenter{\vbox{\moveleft\wd\thr@\boxz}}}%
595         \_math_split_debug_typeout:n {rendsplit/aligncase/centertags}
596         \_math_split_debug_typeout:n {finalize~innertable~aligncase}
597         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}
598       }%
599     \else
600       \_math_split_debug_typeout:n{rendsplit@/aligncase/tbtags}
601       %TODO tbtags is not correct yet
602       \global\setbox7 \hbox{\unhbox\tw@\unskip}%
603       \gdef\split@{%
604         \global\@tempcnta\column@
605         &\setboxz@h{%
606           \savetaglength@
607           \global\advance\row@\@ne
608           \vbox{\moveleft\wd\thr@\box9}%
609         \crrr
610         \noalign{\global\lineht\z@}%
611         \add@amps\@tempcnta
612         \UseTaggingSocket{math/luamml/annotate/false}{\box\thr@}
613         &\box7
614         \_math_split_debug_typeout:n {finalize~innertable~aligncase}
615         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}%
616       }%
617     \fi

```

and again the if we are not in align we need to test for `\gathersplit`

```

618   \else
619     \ifctagsplit@
620       \bool_if:NTF \l__math_gathersplit_bool
621       {
622         \_math_split_debug_typeout:n {rendsplit/equationcase/centertags}
623         \gdef\split@%
624         {\UseTaggingSocket{math/luamml/annotate/false}{\vcenter{\boxz}}}
625       }
626       {
627         \_math_split_debug_typeout:n {rendsplit/gathercase/centertags}

```



```

628         \gdef\split@{\vcenter{\boxz@}%
629         \_math_split_debug_typeof:n {finalize-innertable-gathercase}
630         \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
631     }
632 \else
633     \bool_if:NTF \l__math_gathersplit_bool
634     {
635         \_math_split_debug_typeof:n {rendsplit/equationcase/tbtags}
636         \gdef\split@{%
637             \UseTaggingSocket{math/luamml/annotate/false}{\boxz@}}
638     }
639     {
640         \_math_split_debug_typeof:n {rendsplit/gathercase/tbtags}
641         \gdef\split@{%
642             \boxz@
643             \UseTaggingSocket{math/luamml/mtable/innertable/finalize}}
644     }
645 \fi
646 \fi
647 \aftergroup\split@
648 }

```

(End of definition for `\rendsplit@`. This function is documented on page ??.)

## 2.7 `\intertext`

The `\intertext` command errors with active tagging as it is processed twice which leads to duplicated structures.

```

\intertext@ TODO: review and add sockets!
649 \def\intertext@{%
650     \def\intertext##1{%
651         \ifvmode\else\\\@empty\fi
652         \noalign{%
653             \penalty\postdisplaypenalty\vskip\belowdisplayskip
654             \vbox{
Stop tagging when measuring:
655                 \ifmeasuring@\tag_suspend:n{\measuring}\fi
656                 \normalbaselines
657                 \ifdim\linewidth=\columnwidth
658                 \else \parshape\@ne \@totalleftmargin \linewidth
659                 \fi
End the previous mc:
660                 \tag_mc_end_push:
We are already in a par so we change now to text:
661                 \tagpdfsetup{para/tag=P}%
TODO why \tagpdfpara0n needed?
662                 \tagpdfpara0n
663                 \noindent\ignorespaces##1\par

```

Restart the MC

```
664     \tag_mc_begin_pop:n{}}%
665     \penalty\predisplaypenalty\vskip\abovedisplayskip%
666   }%
667 }}
```

(End of definition for `\intertext@`. This function is documented on page ??.)

## 2.8 `\text`

The `\text` command uses `\mathchoice` which “typesets” the argument four times. This makes it quite problematic for tagging. Without precautions structure objects would be created four times and would get MC-chunks as kids that don’t really exist. `amsmath` contains a switch that allows to execute code only in the first (displaymath) branch, but that isn’t usable here. At first because we don’t know if the first branch creates the same structure as the one that is actually used. At second because the engines executes some commands like `\label` and `\pdfannot` only at shipout from the branch that really was used. So we would get structure data from one `\mathchoice`-branch and MC-labels and links from another one and that gets very messy.

We therefore have to avoid that tagging is active in unused branches. In `pdflatex` it is not possible to detect the `mathstyle` before, so we use a label. With `lualatex` is possible to redefine `\text` not to use `\mathchoice`

`\text@`

```
668 \AddToHook{package/amstext/after}
669 {
currently amsmath is loaded in a begindocument hook, so this test is fine. If amstext is
loaded earlier (in the kernel), this needs perhaps a change.
670   \tag_if_active:T
671   {
672     \sys_if_engine_luatex:TF
673     {
674       \def\text@#1{
675         \tag_socket_use:nnn {math/luamml/hbox}{-}
676         {%
677           \ifcase\mathstyle
678             \hbox{#1}\or
679             \hbox{#1}\or
680             \hbox{#1}\or
681             \hbox{#1}\or
682             \hbox{\let\f@size\sf@size\selectfont#1}\or
683             \hbox{\let\f@size\sf@size\selectfont#1}\or
684             \hbox{\let\f@size\ssf@size\selectfont#1}\or
685             \hbox{\let\f@size\ssf@size\selectfont#1}\or
686             \ERROR
687           \fi
688           \check@mathfonts
689         }}
690     }
691     {
692       \def\text@#1
693       {{
```

```

694         \int_gincr:N\g__math_mathchoice_int
695         \tag_suspend:n{\text@}
696         \mathchoice
697         {
698           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{0}
699           \textdef@{\displaystyle}\f@size{#1}
700         }
701         {
702           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{2}
703           \textdef@{\textstyle}\f@size{\firstchoice@false #1}
704         }
705         {
706           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{4}
707           \textdef@{\textstyle}\sf@size{\firstchoice@false #1}
708         }
709         {
710           \__math_tag_if_mathstyle:en{mathchoice-\int_use:N\g__math_mathchoice_int}{6}
711           \textdef@{\textstyle} \ssf@size{\firstchoice@false #1}
712         }
713         \check@mathfonts
714     }}
715 }
716 }
717 }

```

(End of definition for \text@. This function is documented on page ??.)

## 2.9 \pmb

\pmb prints its argument three times. For tagging we must mark two of occurrences as artifact. For luatex the attributes in the box must be reset, for this we switch to expl3-boxes.

```

\pmb@@
\pmb@
718 \AddToHook{package/amsbsy/after}
719 {
720   \def\pmb@@#1#2#3{\leavevmode\hbox_set:Nn\l_@@_tmpa_box{xxx#3}
721     \dimen@-\box_wd:N\l_@@_tmpa_box
722     \kern-.5\ex@\box_use:N\l_@@_tmpa_box
723     \tag_mc_end:\tag_mc_begin:n{artifact}
724     \tag_mc_reset_box:N\l_@@_tmpa_box
725     \kern\dimen@\kern.25\ex@\raise.4\ex@\box_use:N\l_@@_tmpa_box
726     \kern\dimen@\kern.25\ex@\box_use_drop:N\l_@@_tmpa_box
727     \tag_mc_end:\tag_mc_begin:n{}}
728 }
729 \def\pmb@#1#2{\hbox_set:Nn\l_@@_tmpa_box{\m@th#1{#2}$}
730   \setboxz@h{\m@th#1\mkern.5mu$}\pmbraise@\wdz@
731   \binrel@{#2}
732   \dimen@-\box_wd:N\l_@@_tmpa_box
733   \binrel@@{
734     \mkern-.8mu\box_use:N\l_@@_tmpa_box
735     \tag_mc_end:\tag_mc_begin:n{artifact}
736     \tag_mc_reset_box:N\l_@@_tmpa_box
737     \kern\dimen@\mkern.4mu\raise\pmbraise@\box_use:N\l_@@_tmpa_box

```

```

738     \kern\dimen@\mkern.4mu\box_use_drop:N\l_@@_tmpa_box
739     \tag_mc_end:\tag_mc_begin:n{
740     }
741 }
742 }

```

(End of definition for `\pbm@@` and `\pmb@`. These functions are documented on page ??.)

```

743 \ExplSyntaxOff
744 </kernel>

```

## 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	C
<code>\,</code> . . . . . 406, 444	<code>\chardef</code> . . . . . 153, 218, 256, 288, 495
<code>@@</code> commands:	<code>\columnwidth</code> . . . . . 657
<code>\l_@@_tmpa_box</code> . 720, 721, 722, 724, 725, 726, 729, 732, 734, 736, 737, 738	<code>\cr</code> . . . . . 148
<code>\</code> . . . . . 651	<code>\crr</code> . . . . . 176, 247, 264, 271, 310, 437, 439, 461, 509, 609
A	cs commands:
<code>\abovedisplayskip</code> . . . . . 665	<code>\cs_new:Npn</code> . . . . . 446
<code>\AddToHook</code> . . . . . 668, 718	
<code>\advance</code> . . . . . 137, 287, 297, 334, 607	D
<code>\aftergroup</code> . . . . . 575, 647	<code>\DeclareRobustCommand</code> . . . . . 21
<code>\array</code> . . . . . 403	<code>\def</code> . . . . . 7, 8, 20, 23, 91, 134, 150, 179, 269, 285, 313, 316, 355, 361, 391, 399, 402, 467, 483, 493, 511, 577, 649, 650, 674, 692, 720, 729
<code>\arraystretch</code> . . . . . 402	<code>\displaystyle</code> . . 98, 102, 117, 121, 164, 165, 228, 230, 237, 242, 259, 260, 307, 357, 363, 500, 501, 504, 505, 699
B	<code>\displaywidth</code> . . . . . 160, 295
<code>\baselineskip</code> . . . . . 411	<code>\dp</code> . . . . . 373
<code>\begin</code> . . . . . 204, 252, 482	E
<code>\begingroup</code> . . . . . 320	<code>\edef</code> . . . . . 472
<code>\belowdisplayskip</code> . . . . . 653	<code>\egroup</code> . . . . 26, 29, 183, 195, 273, 275, 388, 396, 441, 442, 463, 464, 488, 489
<code>\bgroup</code> . . . . . 160, 208, 222, 255, 258, 299, 407, 414, 480, 494, 497	<code>\else</code> . . . . . 12, 32, 101, 108, 120, 127, 146, 175, 203, 214, 251, 280, 294, 303, 314, 332, 341, 344, 347, 351, 377, 382, 453, 454, 456, 465, 476, 534, 544, 558, 599, 618, 632, 651, 658
bool commands:	<code>\end</code> . . . . . 482
<code>\bool_if:NTF</code> . . . . . 546, 560, 620, 633	<code>\endaligned</code> . . . . . 267, <u>269</u>
<code>\bool_new:N</code> . . . . . 447	<code>\endgather</code> . . . . . 6, <u>179</u>
<code>\bool_set_true:N</code> . . . . . 478	<code>\endgroup</code> . . . . . 340, 469
bool internal commands:	<code>\endmultiline</code> . . . . . 8
<code>\l_math_gathersplit_bool</code> . . . . . . . . . . <u>447</u> , 478, 546, 560, 620, 633	<code>\endsplit</code> . . . . . 457
<code>\box</code> . . . . . 338, 374, 530, 539, 550, 554, 564, 569, 608, 612, 613	
box commands:	
<code>\box_use:N</code> . . . . . 722, 725, 734, 737	
<code>\box_use_drop:N</code> . . . . . 726, 738	
<code>\box_wd:N</code> . . . . . 721, 732	



<code>\normalbaselines</code> .....	323, 371, 656	<code>\tag_suspend:n</code> .....	7, 655, 695
<code>\normalfont</code> .....	11, 14, 20	<code>\tagpdfparaOn</code> .....	17, 662
<code>\null</code> .....	372, 406	<code>\tagpdfsetup</code> .....	661
<b>O</b>			
<code>\omit</code> .....	139	TeX and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:	
<code>\or</code> ..	678, 679, 680, 681, 682, 683, 684, 685	<code>\@centering</code> .....	171, 302
<b>P</b>			
<code>\par</code> .....	663	<code>\@currenenvir</code> .....	25, 204
<code>\parshape</code> .....	658	<code>\@display@init</code> .....	287
<code>\pdfannot</code> .....	18	<code>\@empty</code> .....	291, 457, 476, 651
<code>\penalty</code> .....	653, 665	<code>\@eqnswtrue</code> .....	146, 454
<code>\pmb</code> .....	19	<code>\@gobble</code> .....	293
<code>\postdisplaypenalty</code> .....	653	<code>\@ifnextchar</code> .....	400
<code>\predisplaypenalty</code> .....	665	<code>\@align</code> .....	96, 115, 142
<code>\ProvidesFile</code> .....	3	<code>\@mathmargin</code> ..	294, 297, 312, 313, 334
<b>Q</b>			
<code>\quad</code> .....	403	<code>\@one</code> .....	38, 43, 68, 73, 137, 159, 218, 256, 287, 372, 373, 374, 495, 514, 518, 584, 607, 658
<b>R</b>			
<code>\raise</code> .....	725, 737	<code>\@enx</code> .....	473, 474
<code>\ref</code> .....	22	<code>\@tempcnta</code> .....	604, 611
<code>\relax</code> .....	209, 302, 338, 366	<code>\@totalleftmargin</code> .....	658
<code>\renewcommand</code> .....	201	<code>\@exp</code> .....	280, 452, 457, 465, 469, 471
<code>\renewenvironment</code> ..	37, 42, 47, 52, 57, 62, 67, 72, 77, 187, 250, 278, 405, 448	<code>\add@amps</code> .....	138, 611
<code>\RequirePackage</code> .....	5	<code>\align@</code> .....	4
<code>\rlap</code> .....	322	<code>\align@preamble</code> .....	91
<b>S</b>			
<code>\scriptsize</code> .....	420, 432	<code>\alignedspace@left</code> .....	207, 254
<code>\scriptstyle</code> .....	418, 430	<code>\alignsep@</code> ....	132, 140, 213, 216, 246
<code>\selectfont</code> .....	682, 683, 684, 685	<code>\ams@return@opt@arg</code> .....	248, 265
<code>\setbox</code> 333, 372, 494, 512, 514, 516, 518, 521, 523, 579, 581, 584, 587, 589, 602		<code>\ams@start@box</code> .....	208, 255
<code>smallmatrix</code> (env.) .....	405	<code>\binrel@</code> .....	731
<code>\span</code> .....	175	<code>\binrel@@</code> .....	733
<code>split</code> (env.) .....	448	<code>\black@</code> .....	24, 181, 193, 487
<code>\split</code> .....	151	<code>\boxz@</code> .....	594, 624, 628, 637, 642
sys commands:		<code>\calc@shift@gather</code> .....	169
<code>\sys_if_engine_luatex:TF</code> .....	672	<code>\check@mathfonts</code> .....	688, 713
<b>T</b>			
<code>\tabskip</code> 112, 132, 158, 171, 175, 233, 246, 294		<code>\collect@body</code> .....	457, 459
<code>\tag</code> .....	152, 290, 293, 496	<code>\column@</code> .....	221, 604
tag commands:		<code>\column@plus</code> .....	223, 234
<code>\tag_if_active:TF</code> .....	670	<code>\common@align@ending</code> .....	
<code>\tag_mc_begin:n</code> ..	723, 727, 735, 739	. 23, 40, 45, 50, 55, 60, 65, 70, 75, 80	
<code>\tag_mc_begin_pop:n</code> .....	664	<code>\default@tag</code> .....	219, 410, 496
<code>\tag_mc_end:</code> .....	723, 727, 735, 739	<code>\df@tag</code> .....	471, 473
<code>\tag_mc_end_push:</code> .....	660	<code>\dimen@</code> .....	334, 335, 336, 338, 721, 725, 726, 732, 737, 738
<code>\tag_mc_reset_box:N</code> .....	724, 736	<code>\displ@y@</code> .....	154
<code>\tag_socket_use:nmn</code> .....	675	<code>\dollar@end</code> 33, 184, 196, 389, 397	
		<code>\dspbrk@context</code> 153, 218, 256, 288, 495	
		<code>\dspbrk@lvl</code> .....	287
		<code>\endmathdisplay@a</code> .....	483
		<code>\endmultline@math</code> ..	10, 307, 355, 366
		<code>\env@cases</code> .....	399
		<code>env@cases</code> .....	399
		<code>\ex@</code> .....	411, 412, 722, 725, 726
		<code>\f@size</code> ..	682, 683, 684, 685, 699, 703
		<code>\firstchoice@false</code> ....	703, 707, 711
		<code>\gather@</code> .....	150, 481

<code>\gather@split</code>	12–15, 459, 467	<code>\reset@equation</code>	469
<code>\gdisplaywidth@</code>	175	<code>\restore@math@cr</code>	256, 289, 409, 495
<code>\gmeasure@</code>	156	<code>\restorealignstate@</code>	28
<code>\gobble@tag</code>	293	<code>\restorecolumn@</code>	274
<code>\if@display</code>	450	<code>\RIfM@</code>	202, 251
<code>\if@eqnsw</code>	136	<code>\row@</code>	137, 159, 287, 607
<code>\if@fleqn</code>	294, 296, 301, 311	<code>\savecolumn@</code>	206
<code>\ifctagsplit@</code>	526, 545, 591, 619	<code>\savefieldlength@</code>	107, 126
<code>\ifinalign@</code>	513, 578	<code>\savetaglength@</code>	606
<code>\ifingather@</code>	27	<code>\set@field</code>	111, 130
<code>\ifmeasuring@</code>	7, 10, 99, 106, 118, 125, 347, 351, 655	<code>\set@gather@field</code>	170
<code>\ifshifftag@</code>	321, 367	<code>\setboxz@h</code>	94, 113, 142, 162, 172, 605, 730
<code>\ifst@rred</code>	135, 146, 454	<code>\sf@size</code>	682, 683, 707
<code>\iftag@</code>	141, 319, 362, 470	<code>\shifftag@false</code>	157
<code>\iftagsleft@</code>	175, 280, 318, 465	<code>\split@</code>	528, 537, 549, 554, 563, 568, 575, 592, 603, 623, 628, 636, 641, 647
<code>\ingather@true</code>	151	<code>\split@aligned</code>	452
<code>\insplit@</code>	14, 151, 493	<code>\split@err</code>	457
<code>\intertext@</code>	154, 649	<code>\split@tag</code>	472, 476, 481
<code>\label@in@display</code>	152	<code>\spread@equation</code>	220, 257, 479
<code>\lendmultline@</code>	280, 391	<code>\ssf@size</code>	684, 685, 711
<code>\lendsplit@</code>	14, 465, 511	<code>\st@rredfalse</code>	38, 48, 58, 68
<code>\Let@</code>	154, 218, 256, 286, 408, 495	<code>\st@rredtrue</code>	43, 53, 63, 73, 78, 189, 278
<code>\lineht@</code>	147, 328, 373, 610	<code>\start@align</code>	38, 43, 48, 53, 58, 63, 68, 73, 78
<code>\m@ne</code>	38, 43, 58, 63, 210, 287	<code>\start@aligned</code>	201
<code>\m@th</code>	11, 14, 20, 98, 117, 164, 227, 236, 259, 307, 417, 429, 500, 504, 729, 730	<code>\start@gather</code>	189
<code>\make@display@tag</code>	142, 326, 333, 375, 379	<code>\start@multline</code>	278
<code>\maketag@@@</code>	2, 8, 22	<code>\strut@</code>	93, 142, 161, 172, 225, 259, 306, 325, 375, 499
<code>\maketag@@@notag</code>	20, 22	<code>\tag@false</code>	291
<code>\maketag@@@notog</code>	20	<code>\tag@in@align</code>	152, 290
<code>\math@cr</code>	24, 180, 192, 386, 394, 486	<code>\tag@true</code>	136, 474
<code>\math@cr@@@</code>	155, 212, 215	<code>\tagform@</code>	2, 22
<code>\math@cr@@@align</code>	134	<code>\text@</code>	668
<code>\math@cr@@@aligned</code>	215	<code>\textdef@</code>	699, 703, 707, 711
<code>\math@cr@@@alignedat</code>	212	<code>\thr@@</code>	523, 529, 530, 538, 539, 589, 593, 594, 608, 612
<code>\math@cr@@@gather</code>	155	<code>\toks@</code>	471, 473
<code>\maxfields@</code>	138, 209, 210, 211	<code>\totwidth@</code>	24, 181, 193, 295, 297, 300, 487
<code>\measuring@true</code>	7	<code>\tw@</code>	58, 63, 78, 211, 521, 587, 602
<code>\mmeasure@</code>	292	<code>\wdz@</code>	730
<code>\multline@</code>	285	<code>\z@</code>	48, 53, 147, 153, 221, 288, 333, 334, 338, 494, 512, 515, 580, 610
<code>\multline@indent</code>	313, 316, 331, 342, 345	<code>\z@skip</code>	112, 158, 175, 213, 233, 294
<code>\new@ifnextchar</code>	400	<code>\text</code>	18
<code>\nonmatherr@</code>	204, 252	<code>\textstyle</code>	703, 707, 711
<code>\pbm@@</code>	718	<code>\textup</code>	22
<code>\place@tag</code>	143	<code>\the</code>	473
<code>\place@tag@gather</code>	173	<code>\thickspace</code>	426
<code>\pmb@</code>	718		
<code>\pmb@@</code>	720		
<code>\pmbraise@</code>	730, 737		
<code>\raise@tag</code>	291, 329, 370		
<code>\rendmultline@</code>	280, 361		
<code>\rendsplit@</code>	14, 15, 465, 577		

U	
<code>\unhbox</code> .....	602
<code>\unhcopy</code> .....	519, 585
<code>\unskip</code> .....	520, 522, 582, 586, 588, 602
<code>\unvcopy</code> .....	512, 515, 580
use commands:	
<code>\use_none:n</code> .....	446
<code>\UseExpandableTaggingSocket</code> .....	25, 182, 194, 272, 387, 395, 440, 462, 490
<code>\UseTaggingSocket</code> .....	13, 15, 17, 102, 109, 121, 128, 145, 165, 168, 174, 207, 230, 232, 242, 244, 254, 260, 262, 276, 348, 352,
	357, 359, 363, 365, 381, 406, 421, 423, 433, 435, 443, 444, 501, 503, 505, 507, 532, 541, 550, 556, 564, 571, 597, 612, 615, 624, 630, 637, 643
V	
<code>\vbox</code> .....	322, 328, 494, 514, 530, 539, 594, 608, 654
<code>\vcenter</code> .....	407, 480, 530, 550, 554, 594, 624, 628
<code>\vskip</code> .....	653, 665
<code>\vtop</code> .....	369, 512, 579
W	
<code>\wd</code> ..	334, 529, 530, 538, 539, 593, 594, 608