

low level

TEX

marks

## Contents

1	Introduction	1
2	The basics	2
3	Migration	3
4	Tracing	5
5	High level commands	6
6	Pitfalls	9

## 1 Introduction

Marks are one of the subsystems of  $\text{T}_{\text{E}}\text{X}$ , as are for instance alignments and math as well as inserts which they share some properties with. Both inserts and marks put signals in the list that later on get intercepted and can be used to access stored information. In the case of inserts this is typeset materials, like footnotes, and in the case of marks it's token lists. Inserts are taken into account when breaking pages, and marks show up when a page has been broken and is presented to the output routine. Marks are used for running headers but other applications are possible.

In MkII marks are used to keep track of colors, transparencies and more properties that work across page boundaries. It permits picking up at the top of a page from where one left at the bottom of the preceding one. When MkII was written there was only one mark so on top of that a multiple mark mechanism was implemented that filtered specific marks from a collection. Later,  $\varepsilon\text{-T}_{\text{E}}\text{X}$  provided mark classes so that mechanism could be simplified. Although it is not that hard to do, this extension to  $\text{T}_{\text{E}}\text{X}$  didn't add any further features, so we can assume that there was no real demand for that.<sup>1</sup>

But, marks have some nasty limitations, so from the  $\text{ConT}_{\text{E}}\text{Xt}$  perspective there always was something to wish for. When you hide marks in boxes they will not be seen (the same is true for inserts). You cannot really reset them either. Okay, you can set them to nothing, but even then already present marks are still there. The  $\text{LuaT}_{\text{E}}\text{X}$  engine has a `\clearmarks` primitive but that works global. In  $\text{LuaMetaT}_{\text{E}}\text{X}$  a proper mark flusher is available. That engine also can work around the deeply nested disappearing marks. In addition, the current state of a mark can be queried and we have some tracing facilities.

In MkIV the engine's marks were not used at all and an alternative mechanism was written using Lua. It actually is one of the older MkIV features. It doesn't have the side

---

<sup>1</sup> This is probably true for most  $\text{LuaT}_{\text{E}}\text{X}$  and  $\text{LuaMetaT}_{\text{E}}\text{X}$  extensions, maybe example usage create retrospective demand. But one reason for picking up on engine development is that in the  $\text{ConT}_{\text{E}}\text{Xt}$  perspective we actually had some demands.

effects that native marks have but it comes at the price of more overhead, although that is bearable.

In this document we discuss marks but assume that LuaMetaT<sub>E</sub>X is used with ConT<sub>E</sub>Xt LMTX. There we experiment with using the native marks, complemented by a few Lua mechanisms, but it is to be seen if that will be either a replacement or an alternative.

## 2 The basics

Although the original T<sub>E</sub>X primitives are there, the plural  $\varepsilon$ -T<sub>E</sub>X mark commands are to be used. Marks, signals with token lists, are set with:

```
\marks0{This is mark 0} % equivalent to: \mark{This is mark 0}
\marks4{This is mark 4}
```

When a page has been split off, you can (normally this only makes sense in the output routine) access marks with:

```
\topmarks 4
\firstmarks4
\botmarks 4
```

A ‘top’ mark is the last one on the previous page(s), the ‘first’ and ‘bottom’ refer to the current page. A mark is a so called node, something that ends up in the current list and the token list is stored with it. The accessors are just commands and they fetch the token list from a separately managed storage. When you set or access a mark that has not yet been used, the storage is bumped to the right size, so it doesn't make sense to use e.g. `\marks 999` when there are no 998 ones too: it not only takes memory, it also makes T<sub>E</sub>X run over all these mark stores when synchronization happens. The best way to make sure that you are sparse is:

```
\newmarks\MyMark
```

Currently the first 16 marks are skipped so this makes `\MyMark` become mark 17. The reason is that we want to make sure that users who experiment with marks have some scratch marks available and don't overload system defined ones. Future versions of ConT<sub>E</sub>Xt might become more restrictive.

Marks can be cleared with:

```
\clearmarks 4
```

which clears the storage that keeps the top, first and bot marks. This happens immediately. You can delay this by putting a signal in the list:

`\flushmarks` 4

This (LuaMetaT<sub>E</sub>X) feature makes it for instance easy to reset marks that keep track of section (and lower) titles when a new chapter starts. Of course it still means that one has to implement some mechanism that deals with this but ConT<sub>E</sub>Xt always had that.

The current, latest assigned, value of a mark is available too:

`\currentmarks` 4

Using this value in for instance headers and footers makes no sense because the last node set can be on a following page.

### 3 Migration

In the introduction we mentioned that LuaMetaT<sub>E</sub>X has migration built in. In MkIV we have this as option too, but there it is delegated to Lua. It permits deeply nested inserts (notes) and marks (but we don't use native marks in MkIV).

Migrated marks end up in the postmigrated sublist of a box. In other lowlevel manuals we discuss these pre- and postmigrated sublists. As example we use this definition:

```
\setbox0\vbox\bgroup
test \marks 4 {mark 4.1}\par
test \marks 4 {mark 4.1}\par
test \marks 4 {mark 4.1}\par
\egroup
```

When we turn migration on (officially the second bit):

`\automigrationmode"FF \showbox0`

we get this:

```
> \box0=
2:4: \vbox[normal][...], width 483.69687, height 63.43475, depth 0.15576, direction l2r
2:4: ..\list
2:4: ... \hbox[line][...], width 483.69687, height 7.48193, depth 0.15576, glue 459.20468fil, direction l2r
2:4: ... \list
2:4: .... \glue[left hang][...] 0.0pt
2:4: .... \glue[left][...] 0.0pt
2:4: .... \glue[parfillskip][...] 0.0pt
2:4: .... \par[newgraf][...], hangafter 1, hsize 483.69687, pretolerance 100, tolerance 200, adjdemerits 10000, linepenalty 10, doublehyphenemerits 10000,
    finalhyphenemerits 5000, clubpenalty 2000, widowpenalty 2000, brokenpenalty 100, parfillskip 0.0pt plus 1.0fil, hyphenationmode 499519
2:4: .... \glue[indent][...] 0.0pt
2:4: .... \glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: .... \glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000065 e
2:4: .... \glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000073 s
```

```

2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\glue[space][...] 3.49658pt plus 1.74829pt minus 1.16553pt, font 8
2:4: ....\penalty[line][...] 10000
2:4: ....\glue[parfill][...] 0.0pt plus 1.0fil
2:4: ....\glue[right][...] 0.0pt
2:4: ....\glue[right hang][...] 0.0pt
2:4: ....\glue[par][...] 11.98988pt plus 3.99663pt minus 3.99663pt
2:4: ....\glue[baseline][...] 8.34883pt
2:4: ....\hbox[line][...], width 483.69687, height 7.48193, depth 0.15576, glue 459.20468fil, direction l2r
2:4: ...list
2:4: ....\glue[left hang][...] 0.0pt
2:4: ....\glue[left][...] 0.0pt
2:4: ....\glue[parfillleft][...] 0.0pt
2:4: ....\par[newgraf][...], hangafter 1, hsize 483.69687, pretolerance 100, tolerance 200, adjdemerits 10000, linepenalty 10, doublehyphenemerits 10000,
    finalhyphenemerits 5000, clubpenalty 2000, widowpenalty 2000, brokenpenalty 100, parfillskip 0.0pt plus 1.0fil, hyphenationmode 499519
2:4: ....\glue[indent][...] 0.0pt
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000065 e
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000073 s
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\glue[space][...] 3.49658pt plus 1.74829pt minus 1.16553pt, font 8
2:4: ....\penalty[line][...] 10000
2:4: ....\glue[parfill][...] 0.0pt plus 1.0fil
2:4: ....\glue[right][...] 0.0pt
2:4: ....\glue[right hang][...] 0.0pt
2:4: ....\glue[par][...] 11.98988pt plus 3.99663pt minus 3.99663pt
2:4: ....\glue[baseline][...] 8.34883pt
2:4: ....\hbox[line][...], width 483.69687, height 7.48193, depth 0.15576, glue 459.20468fil, direction l2r
2:4: ...list
2:4: ....\glue[left hang][...] 0.0pt
2:4: ....\glue[left][...] 0.0pt
2:4: ....\glue[parfillleft][...] 0.0pt
2:4: ....\par[newgraf][...], hangafter 1, hsize 483.69687, pretolerance 100, tolerance 200, adjdemerits 10000, linepenalty 10, doublehyphenemerits 10000,
    finalhyphenemerits 5000, clubpenalty 2000, widowpenalty 2000, brokenpenalty 100, parfillskip 0.0pt plus 1.0fil, hyphenationmode 499519
2:4: ....\glue[indent][...] 0.0pt
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000065 e
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000073 s
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\glue[space][...] 3.49658pt plus 1.74829pt minus 1.16553pt, font 8
2:4: ....\penalty[line][...] 10000
2:4: ....\glue[parfill][...] 0.0pt plus 1.0fil
2:4: ....\glue[right][...] 0.0pt
2:4: ....\glue[right hang][...] 0.0pt
2:4: .\postmigrated
2:4: .\mark[4][...]
2:4: ..{mark 4.1}
2:4: .\mark[4][...]
2:4: ..{mark 4.1}
2:4: .\mark[4][...]
2:4: ..{mark 4.1}

```

When we don't migrate, enforced with:

**\automigrationmode"00 \showbox0**

the result is:

```

> \box0=
2:4: \vbox[normal][...], width 483.69687, height 63.43475, depth 0.15576, direction l2r
2:4: .list
2:4: ..\hbox[line][...], width 483.69687, height 7.48193, depth 0.15576, glue 459.20468fil, direction l2r
2:4: ...list
2:4: ....\glue[left hang][...] 0.0pt
2:4: ....\glue[left][...] 0.0pt
2:4: ....\glue[parfillleft][...] 0.0pt
2:4: ....\par[newgraf][...], hangafter 1, hsize 483.69687, pretolerance 100, tolerance 200, adjdemerits 10000, linepenalty 10, doublehyphenemerits 10000,
    finalhyphenemerits 5000, clubpenalty 2000, widowpenalty 2000, brokenpenalty 100, parfillskip 0.0pt plus 1.0fil, hyphenationmode 499519
2:4: ....\glue[indent][...] 0.0pt
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000065 e
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000073 s
2:4: ....\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\glue[space][...] 3.49658pt plus 1.74829pt minus 1.16553pt, font 8
2:4: ....\penalty[line][...] 10000
2:4: ....\glue[parfill][...] 0.0pt plus 1.0fil
2:4: ....\glue[right][...] 0.0pt
2:4: ....\glue[right hang][...] 0.0pt
2:4: ..\mark[4][...]
2:4: ..{mark 4.1}
2:4: ..\glue[par][...] 11.98988pt plus 3.99663pt minus 3.99663pt
2:4: ..\glue[baseline][...] 8.34883pt
2:4: ..\hbox[line][...], width 483.69687, height 7.48193, depth 0.15576, glue 459.20468fil, direction l2r

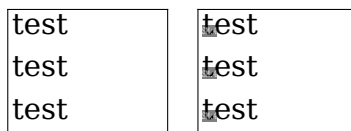
```

```

2:4: ...\\list
2:4: ....\\glue[left hang][...] 0.0pt
2:4: ....\\glue[left][...] 0.0pt
2:4: ....\\glue[parfillleft][...] 0.0pt
2:4: ....\\par[newgraf][...], hangafter 1, hsize 483.69687, pretolerance 100, tolerance 200, adjdemerits 10000, linepenalty 10, doublehyphendemerits 10000,
    finalhyphendemerits 5000, clubpenalty 2000, widowpenalty 2000, brokenpenalty 100, parfillskip 0.0pt plus 1.0fil, hyphenationmode 499519
2:4: ....\\glue[indent][...] 0.0pt
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000065 e
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000073 s
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\\glue[space][...] 3.49658pt plus 1.74829pt minus 1.16553pt, font 8
2:4: ....\\penalty[line][...] 10000
2:4: ....\\glue[parfilll][...] 0.0pt plus 1.0fil
2:4: ....\\glue[right][...] 0.0pt
2:4: ....\\glue[right hang][...] 0.0pt
2:4: ...\\mark[4][...]
2:4: ...{mark 4.1}
2:4: ...\\glue[par][...] 11.98988pt plus 3.99663pt minus 3.99663pt
2:4: ...\\glue[baseline][...] 8.34883pt
2:4: ...\\hbox[line][...], width 483.69687, height 7.48193, depth 0.15576, glue 459.20468fil, direction l2r
2:4: ...\\list
2:4: ....\\glue[left hang][...] 0.0pt
2:4: ....\\glue[left][...] 0.0pt
2:4: ....\\glue[parfillleft][...] 0.0pt
2:4: ....\\par[newgraf][...], hangafter 1, hsize 483.69687, pretolerance 100, tolerance 200, adjdemerits 10000, linepenalty 10, doublehyphendemerits 10000,
    finalhyphendemerits 5000, clubpenalty 2000, widowpenalty 2000, brokenpenalty 100, parfillskip 0.0pt plus 1.0fil, hyphenationmode 499519
2:4: ....\\glue[indent][...] 0.0pt
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000065 e
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000073 s
2:4: ....\\glyph[32768][...], language (n=1,l=2,r=3), hyphenationmode "79F3F, options "80, font <8: DejaVuSerif @ 11.0pt>, glyph U+000074 t
2:4: ....\\glue[space][...] 3.49658pt plus 1.74829pt minus 1.16553pt, font 8
2:4: ....\\penalty[line][...] 10000
2:4: ....\\glue[parfilll][...] 0.0pt plus 1.0fil
2:4: ....\\glue[right][...] 0.0pt
2:4: ....\\glue[right hang][...] 0.0pt
2:4: ...\\mark[4][...]
2:4: ...{mark 4.1}

```

When you say `\showmakeup` or in this case `\showmakeup[mark]` the marks are visualized:



enabled

disabled

Here `sm` means ‘set mark’ while `rm` would indicate a ‘reset mark’. Of course migrated marks don't show up because these are bound to the box and thereby have become a a specific box property as can be seen in the above trace.

## 4 Tracing

The LuaMetaTeX engine has a dedicated tracing option for marks. The fact that the traditional engine doesn't have this can be seen as indication that this is seldom needed.

`\tracingmarks1`

`\tracingonline2`

When tracing is set to 1 we get a list of marks for the just split of page:

```

2:7: <mark class 51, top := bot>
2:7: ..{sample 9.1}
2:7: <mark class 51: first := mark>
2:7: ..{sample 10.1}
2:7: <mark class 51: bot := mark>
2:7: ..{sample 10.1}
2:7: <mark class 51, page state>
2:7: ..top {sample 9.1}
2:7: ..first {sample 10.1}
2:7: ..bot {sample 10.1}

```

When tracing is set to 2 you also get details we get a list of marks of the analysis:

```

1:9: <mark class 51, top := bot>
1:9: ..{sample 5.1}
1:9: <mark class 51: first := mark>
1:9: ..{sample 6.1}
1:9: <mark class 51: bot := mark>
1:9: ..{sample 6.1}
1:9: <mark class 51: bot := mark>
1:9: ..{sample 7.1}
1:9: <mark class 51: bot := mark>
1:9: ..{sample 8.1}
1:9: <mark class 51: bot := mark>
1:9: ..{sample 9.1}
1:9: <mark class 51, page state>
1:9: ..top {sample 5.1}
1:9: ..first {sample 6.1}
1:9: ..bot {sample 9.1}

```

## 5 High level commands

I think that not that many users define their own marks. They are useful for showing section related titles in headers and footers but the implementation of that is hidden. The native mark references are `top`, `first` and `bottom` but in the `ConTEXt` interface we use different keywords.

<b>ConT<sub>E</sub>Xt</b>	<b>T<sub>E</sub>X</b>	<b>column</b>	<b>page</b>
previous	top	last before sync	last on previous page
top	first	first in sync	first on page

bottom	bot	last in sync	last on page
first	top	first not top in sync	first on page
last	bot	last not bottom in sync	last on page
default		the same as first	
current		the last set value	

In order to separate marks in ConT<sub>E</sub>Xt from those in T<sub>E</sub>X, the term ‘marking’ is used. In MkIV the regular marks mechanism is of course there but, as mentioned, not used. By using a different namespace we could make the transition from MkII to MkIV (the same is true for some more mechanisms).

A marking is defined with

```
\definemarking[MyMark]
```

A defined marking can be set with two equivalent commands:

```
\setmarking[MyMark]{content}
\marking [MyMark]{content}
```

The content is not typeset but stored as token list. In the sectioning mechanism that uses markings we don't even store titles, we store a reference to a title. In order to use that (deep down) we hook in a filter command. By default that command does nothing:

```
\setupmarking[MyMark][filtercommand=\firstofoneargument]
```

The token list does *not* get expanded by default, unless you set it up:

```
\setupmarking[MyMark][expansion=yes]
```

The current state of a marking can be cleared with:

```
\clearmarking[MyMark]
```

but because that en is not synchronized the real deal is:

```
\resetmarking[MyMark]
```

Be aware that it introduces a node in the list. You can test if a marking is defined with (as usual) a test macro. Contrary to (most) other test macros this one is fully expandable:

```
\doifelsemarking {MyMark} {
  defined
} {
```



undefined

}

Because there can be a chain involved, we can relate markings. Think of sections below chapters and subsections below sections:

`\relatemarking[MyMark][YourMark]`

When a marking is set its relatives are also reset, so setting YourMark will reset MyMark. It is this kind of features that made for marks being wrapped into high level commands very early in the ConT<sub>E</sub>Xt development (and one can even argue that this is why a package like ConT<sub>E</sub>Xt exists in the first place).

The rest of the (relatively small) repertoire of commands has to do with fetching markings. The general command is `\getmarking` that takes two or three arguments:

`\getmarking[MyMarking][first]`  
`\getmarking[MyMarking][page][first]`  
`\getmarking[MyMarking][page][first]`  
`\getmarking[MyMarking][column:1][first]`

There are (normally) three marks that can be fetched so we have three commands that do just that:

`\fetchonemark [MyMarking][which one]`  
`\fetchtwomarks[MyMarking]`  
`\fetchallmarks[MyMarking]`

You can setup a separator key which by default is:

`\setupmarking[MyMarking][separator=\space\emdash\space]`

Injection is enabled by default due to this default:

`\setupmarking[MyMarking][state=start]`

The following three variants are (what is called) fully expandable:

`\fetchonemarking [MyMarking][which one]`  
`\fetchtwomarkings[MyMarking]`  
`\fetchallmarkings[MyMarking]`

## 6 Pitfalls

The main pitfall is that a (re)setting a mark will inject a node which in vertical mode can interfere with spacing. In for instance section commands we wrap them with the title so there it should work out okay.