# emarks[*]

## ε-TEX named marks registers

FC

2011/03/26 – version 1.0

**Abstract**

ε-TEX defines 32 768 marks registers while TEX provided only one !

So small, this package provides commands to access ε-TEX marks registers by their name rather than by their number. This makes the use of them far more comfortable than "old LATEX" tricks with `\markright`, `\markboth` *etc.*

emarks requires ε-TEX and the generic package `etex.sty` for allocation.

Presently designed to be loaded by LATEX, a `plain` TEX version might be provided later...

## Contents of emarks

## List of the listings / examples

To be done !

# 1 The ε-TEX marks registers

`\marksthe{⟨named-mark⟩}{⟨content⟩}` `\marksthecs{⟨named-mark⟩}{⟨cs-name⟩}`
`\marksthe*{⟨named-mark⟩}{⟨content⟩}` `\marksthecs*{⟨named-mark⟩}{⟨cs-name⟩}`

`\marksthe{section}{⟨content⟩}` Marks the ⟨*content*⟩ into the named mark register ⟨*section*⟩ in the same way as the ε-TEX primitive `\marks`: in particular the ⟨*content*⟩ is immediately expanded.
If the mark register does not exist, it is created (or allocated) with `\newmarks` (in `etex.sty`).

`\marksthe*{section}{⟨content⟩}` does the same but the ⟨*content*⟩ is not expanded. The current values of counters, `\thesection` *etc.* will be wrong: they will expand to the value they have at the time the mark register is read, not at the time of `\marksthe*`.
Yet `\marksthe*` is useful to mark a title only like in

```
\def\sectionmark#1{\marksthe*{section}{⟨#1⟩}}
```

or to control the expansion (the ⟨*content*⟩ can be expanded before marking in a way and with the protections desired by the user).

---

[*] This documentation is produced with the DocStrip utility.

| | | |
|---|---|---|
| ⟶ To get the package, | run: | `etex emarks.dtx` |
| ⟶ To get the documentation | run (thrice): | `pdflatex emarks.dtx` |
| To get the index, | run: | `makeindex -s gind.ist emarks.idx` |

The `.dtx` file is embedded into this `.pdf` file thank to embedfile by H. Oberdiek.

Similarly `\marksthecs{⟨subsubsection⟩}{⟨cs-name⟩}` marks the content of `\cs-name` by the mean of the named mark register ⟨*subsubsection*⟩. ⟨*cs-name*⟩ is really the *name of the control sequence* and not the control sequence itself: it does not start with `\`.

If `\cs-name` is empty the mark is empty, but if it is undefined or `\relax`: nothing is marked: at reading time, the mark register never expands to `\undefined` nor to `\relax`.

The syntax follows ε-TEX `\marks` primitive (a token-like syntax): braces are mandatory around the `{⟨content⟩}` to be marked, even if it is made of one single token.

`\thefirstmarks` `{⟨named-mark⟩}`expandable
`\thebotmarks` `{⟨named-mark⟩}`expandable
`\thetopmarks` `{⟨named-mark⟩}`expandable

Those commands are expandable in exactly one step of expansion. If the ⟨*named-mark*⟩ mark register does not exists, the expansion is null (*ie.* nothing is done nor printed).

`\thefirstmarks{⟨chapter⟩}` expands to the content of the first invocation of `\marksthe{⟨chapter⟩}` on the current page if `\marksthe{chapter}` was used on the current page, or the last invocation of `\marksthe{chapter}` if no marks occured on the current page.
TEXnically this is `\firstmarks\marks@chapter`

`\thebotmarks{⟨chapter⟩}` expands to the content of the last invocation of `\marksthe{⟨chapter⟩}` (the most recent `\marks`).
TEXnically this is `\botmarks\marks@chapter`

`\thetopmarks{⟨chapter⟩}` expands to the content of `\botmarks` at the time TEX shipped out the last page.
TEXnically this is `\topmarks\marks@chapter`

`\getthemarks\firstmarks|\botmarks|\topmarks{⟨named-mark⟩} {\control-sequence}`
`\getthefirstmarks{⟨named-mark⟩} {\control-sequence}`
`\getthebotmarks{⟨named-mark⟩} {\control-sequence}`
`\getthetopmarks{⟨named-mark⟩} {\control-sequence}`

`\thefirstmarks`, `\thebotmarks` and `\thetopmarks` expand the content of the mark. To get it in a macro `\getthemarks` can be used: `\control-sequence` is defined as a parameterless macro whose replacement text is the content of the given mark register.

If the ⟨*named-mark*⟩ mark register does not exist, the meaning of `\control-sequence` is undefined.

`\ifmarksvoid{\firstmarks}{named-mark}{⟨true⟩}{⟨false⟩}`
`\ifmarksvoid{\botmarks}{named-mark}{⟨true⟩}{⟨false⟩}`
`\ifmarksvoid{\topmarks}{named-mark}{⟨true⟩}{⟨false⟩}`

`\ifmarksvoid` expands the `{⟨true⟩}` part if either:

- The requested mark register is empty,
- The requested mark register is `\undefined`,
- The requested mark register is `\relax`,
- The ⟨*named-mark*⟩ mark register does not exist.

```
\ifmarksequal{\firstmarks}{\topmarks}{named-mark}{⟨true⟩}{⟨false⟩}
\ifmarksequal{\firstmarks}{\botmarks}{named-mark}{⟨true⟩}{⟨false⟩}
```

Pretty often we want to compare the botmarks against the firstmarks or the topmarks, to adapt the header and/or footer in case those marks are equal or different, *ie.* in case the page contains a new section title or not:

`\ifmarksequal` expands the code in the `{⟨true⟩}` or the `{⟨false⟩}` part if the extraction of the marks are equal (in the sense of `\ifx`) or different.

If any of the marks register `\marks@named-mark` does not exist the `{⟨false⟩}` part is expanded.

If marks are used both at `\sectionmark` **and at** `\sectionbreak` then the following assertions are true:

- `\firstmarks` = `\botmarks` ⇔ there is at most one section title on the current page;
- `\topmarks` = `\botmarks` ⇔ there is no section title on the current page;
- `\firstmarks` = `\topmarks` ⇔ the last section title continues on the current page.

```
\showthemarks{⟨named-mark⟩}
```

`\showthemarks` is for debugging purpose: it prints a message in the .log file and the "standard error" with the contents of the marks `\firstmarks`, `\botmarks` and `\topmarks` for the ⟨*named-mark*⟩ register given. Then it executes `\show` on the extracted content of `\firstmarks` in order to stop compilation at that point: the console displays the contents of `\firstmarks`, `\botmarks` and `\topmarks`.

emarks

# 2 IMPLEMENTATION

### Identification

The package namespace is **\em@rks**

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}[2005/12/01]
\ProvidesPackage{emarks}
         [2011/03/26 v1.0 - e-TeX named marks registers (FC)]
\RequirePackage {etex}
```

**\emarks@newmarks** allocates a new marks register if it does not exists.

```
\def\emarks@newmarks #1{\PackageInfo {emarks}{New marks register `#1'}%
                        \newmarks #1% \newmarks is global !!
}% \emarks@newmarks
```

**\marksthe**
**\marksthecs**

```
\marksthe  { named−mark }{ general text }
\marksthe* { named−mark }{ general text }
\marksthe  { named−mark }{ named control sequence }
\marksthecs*{ named−mark }{ named control sequence }
```

```
\protected\def\marksthe   {\emarks@setmarks {}}
\protected\def\marksthecs {\emarks@setmarks {\toks@\expandafter{\csname\the\toks@\endcsname}}
\def\emarks@setmarks #1{\begingroup \@ifstar {\emarks@ {#1}\def  }
                                             {\emarks@ {#1}\edef }%
}% \emarks@setmarks
\def\emarks@ #1#2#3{\def\@tempa
      {#1#2\@tempa {\the\toks@ }\expandafter\emarks@marks \csname marks@#3\endcsname }%
                                                \afterassignment \@tempa \toks@ =
}% \emarks@
\def\emarks@marks #1{\ifx \relax#1\emarks@newmarks #1\fi \marks #1{\@tempa }\endgroup }
```

**\thefirstmarks**
**\thebotmarks**
**\thetopmarks**

\thefirstmarks extract the **\firstmarks** from a named mark register. The macros are purely expandable in exactly one step of expansion.

```
\newcommand*\thefirstmarks {\romannumeral \emarks@themarks \firstmarks  }
\newcommand*\thebotmarks   {\romannumeral \emarks@themarks \botmarks    }
\newcommand*\thetopmarks   {\romannumeral \emarks@themarks \topmarks    }
\def\emarks@themarks #1#2{\expandafter \ifx
    \csname\ifcsname marks@#2\endcsname marks@#2\else relax\fi\endcsname\relax
            \expandafter \z@
    \else   \expandafter \z@ #1\csname marks@#2\expandafter \endcsname  \fi
}% \emarks@themarks
```

**\getthemarks**
**\getthefirstmarks**
**\getthebotmarks**
**\getthetopmarks**

Extract the marks and store in a parameterless macro.

```
\protected\def\getthemarks #1#2#3{\ifcsname marks@#2\endcsname
          \expandafter \def \expandafter #3\expandafter {#1\csname marks@#2\endcsname}%
    \else                           \let #3=\@undefined    \fi
}% \getthemarks
\protected\def\getthefirstmarks {\getthemarks   \firstmarks }
\protected\def\getthebotmarks   {\getthemarks   \botmarks   }
\protected\def\getthetopmarks   {\getthemarks   \topmarks   }
```

**\ifmarksvoid** Test if a marks is defined, not empty and not **\relax**.

```
\protected\def\ifmarksvoid #1#2{\begingroup \getthemarks {#1}{#2}\x
    \ifodd \ifdefined\x \ifx \x\relax 0 \fi \ifx \x\@empty 0 \fi \else 0 \fi
           1 \endgroup\expandafter\@secondoftwo
    \else    \endgroup\expandafter\@firstoftwo      \fi
}% \ifmarksvoid
```

**\ifmarksequal** Test with **\ifx** if two marks are equal:

```
\ifmarksequal \firstmarks \botmarks { named-mark }
```

```
\protected\def\ifmarksequal #1#2#3{\begingroup \getthemarks{#1}{#3}\x \getthemarks{#2}{#3}\y
        \expandafter \endgroup \ifodd \ifdefined\x \ifdefined\y \ifx \x\y 0 \fi\fi\fi
                                      1 \expandafter\@secondoftwo
                               \else    \expandafter\@firstoftwo      \fi
}% \ifmarksequal
```

**\showthemarks** Shows the contents of the marks registers

```
\protected\def\showthemarks #1{\begingroup   \emarks@showthemarks 0{#1}\firstmarks
                                            \emarks@showthemarks 2{#1}\botmarks
                                            \emarks@showthemarks 4{#1}\topmarks
    \message{firstmarks "#1": \the\toks0^^J%
             botmarks   "#1": \the\toks2^^J%
             topmarks   "#1": \the\toks4^^J}\show\@tempa
    \endgroup
}% \showthemarks
\def\emarks@showthemarks #1#2#3{\getthemarks #3{#2}\@tempa \toks #1 = \ifdefined\@tempa
    \expandafter\ifx \noexpand\@tempa\@tempa {}\else \expandafter {\@tempa }\fi
                                                                      \else {}\fi
}% \emarks@showthemarks
⟨/package⟩
```

## 3 History

[2011/03/26 v1.0]

- First version.

## 4 References

[1] The etex package by Peter Breitenlohner
v2.0 eTeX basic definition package (PEB)
CTAN:help/Catalogue/entries/etex-pkg.html

## 5 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.