

Editorial Notes for L^AT_EX*

Michael Kohlhase
Computer Science, Jacobs University
<http://kwarc.info/kohlhase>

December 14, 2010

Abstract

This package defines a couple of editorial notes that simplify collaboration on a L^AT_EX text. These allow authors to annotate status information in the source. In draft mode, the annotations are shown for communication, and in publication mode these are suppressed.

1 Introduction

When collaborating on a document with multiple authors, communication about the status of a given passage and intentions about what to do with it, indications about issues still need to be resolved, and directives to other authors, e.g. calling for help, or passing the baton, etc. make up much of the overhead involved in collaboration. In particular, many of the necessary communicative acts are centered around specific points or passages in the document at hand. Therefore it is natural to embed the communicative acts in the document source itself. The simplest version of this is to special markers like “(*** remember to rework this before publication ***)”, where the markers “(***” and “***)” serve as a visual aid and target for search and navigation tools in the editor. Of course this is dangerous, and we have all seen published texts with such markers still present.

The L^AT_EX package described in here systematizes the idea and provides more conspicuous visual markers (as footnotes and margin notes) and a way of making all all of these private markers and comments invisible for publication or outside communication.

2 The User Interface

As usual in L^AT_EX, the package is loaded by `\usepackage[(options)]{ed}`, where [*(options)*] is optional and gives a comma separated list of options. The `ed` package takes the options `show` and `hide`, where `hide` is the default case, so that `\usepackage{ed}`, is equivalent to `\usepackage[hide]{ed}`. If the `show` option is given, then the editorial notes are presented as special annotations to the document, otherwise they are completely invisible — if you take care about spaces in the source code. For instance `text\ednote{...}_text` will fool T_EX’s whitespace-collapsing algorithm and bring it to output two spaces in the document instead of just one as expected `text\ednote{...}_text` should be used instead! This `hide` option useful for preparing “clean” version for outside consumption without losing the management metadata. In some situations the `\marginpar` decorations generated by editorial notes are disallowed, for these situations the `ed` package has the `nomargins` option that disables them.

As many classes support the `draft` and `final` options and pass them on to all loaded packages that will take them, the `ed` package supports them as aliases for `show` and `hide`.

*Version v1.7 (last revised 2011/01/16)

2.1 Annotation Macros

The main user macro of this package is `\ednote`. It is used say what you have done or what should still be done at a given point in the document. `ed.sty` formats it like a footnote, but with a margin note that marks the place in the text, where the note is located. Otherwise, in the presence of multiple `\ednotes` in a page it may be difficult to find the referenced locations¹. Editorial notes are numbered and marked in the margin for easy recognition. `\ednote` also takes an optional argument that is an identifier. This allows to cross-reference ednotes in each other.

`\ednote`

EdNote:1

The `\edissue` macro² is a variant of `\ednote` for issues that still have to be discussed among the authors. For issues the labeling and cross-referencing in the optional argument is especially useful. Note that the `\issue` macro which earlier versions of the `ed` package provided is deprecated in favor of `\edissue`. The old `\issue` macro is only provided for backwards compatibility if it is not defined by other packages. For this to work, the `ed` package should be loaded late in the preamble.

`\edissue`
Issue:2

`\issue`

The `\tweak` macro³ is a variant of `\ednote` for marking places where we have tweaked something (apart from the standard way we usually use).

`\tweak`
Tweak:3

The `todo` environment is an environment that can be used to mark up writing tasks. These are inserted into the text in a different font, so that they are conspicuous as a foreign part, and take an argument for a comment.

`todo`

The `todolist` is a variant of the `todo` environment which is a is an itemized list.

`todolist`

The following is an example, generated from the the input `\begin{todolist}{an example todolist} ... \item ... \end{todolist}`. To Do: an example todolist

ToDo

1. lots of good examples
2. a general chapter about best practices

the `newpart` environment can be used to mark up changed text blocks. `\begin{newpart}` takes an argument that is interpreted as a comment and is treated like an `\ednote` comment. The `oldpart` environment is similar to `newpart` but is used for old parts of text copied from another document that still need to be changed in a document. They displayed in gray, so that they can be identified better.

`newpart`

`oldpart`

The annotation macros have capitalized variants (`\Ednote`, `\edIssue`, `\Tweak`, `Todo`, `Newpart`, `Oldpart`) that do not make location marks in the margin. This is convenient in situations (e.g. inside boxes or minipages) that allow footnotes, but no margin notes.

2.2 Generating Statistics and Explanations

Up to version 1.6, the `ed` package had an explicit macro `\ednotemessage` that put just before the `\end{document}` will generates a message with cardinality information for the ednotes into the log file. This macro is now deprecated, since the message is now automatically generated.

`\ednotemessage`

The `\edexplanation` macro generates an explanation of the best practices into the document. So that if you introduce collaboration partners to the `ed` package, you can give them an explanation “in-place”.

`\edexplanation`

2.3 Configuration

The font shape of editorial annotations is governed by the parameter `\ednoteshape` the default is sans serif, specialize it to say italic by `\def\ednoteshape{\it}`. The labels in the margins can be tweaked (e.g. for localization:) by setting the (internal) macros `\ednote@label`, `\tweak@label`, `\edissue@label`, `\b@todo@label`, `\e@todo@label`.

`\ednoteshape`

¹EDNOTE: this is an example of an ednote

²ISSUE: The control sequence `issue` conflicts with many other macro packages, maybe we should rename it to `edissue`.

³TWEAK: this is an example of an tweak

2.4 Best Practices

In a situation with multiple authors, we it is advisable to use the `\ednote` in the following form:
`\ednote{author: some explanatory text}` to specify the originator of the note.

3 The Implementation

The implementation is rather standard. We first set up the options for the package. The main switch is `\showednotes`, which governs the visibility of the annotations.

```
1 (*package)
2 \newif\ifshowednotes\showednotesfalse
3 \newif\ifmargins\marginstrue
```

the next step is to declare the package options, they just set `\showednotes` switch accordingly.

```
4 \DeclareOption{show}{\showednotestrue\message{ed.sty: showing ednotes}}
5 \DeclareOption{hide}{\showednotesfalse\message{ed.sty: hiding ednotes}}
6 \DeclareOption{draft}{\showednotestrue\message{ed.sty: showing ednotes}}
7 \DeclareOption{final}{\showednotesfalse\message{ed.sty: hiding ednotes}}
8 \DeclareOption{nomargins}{\marginsfalse}
9 \ProcessOptions
```

The next step is to load the `verbatim` or `paralist` packages, so that we can either comment out the or use the `compactenum` environment for todo lists.

```
10 \ifshowednotes
11 \RequirePackage{paralist}
12 \RequirePackage{xcolor}
13 \else
14 \RequirePackage{verbatim}
15 \fi
```

This ends the package setup code, so we can come to the implementation of the functionality of the package.

`\ednoteshape` We start with the configuration part, predefining `\epdnoteshape` to be sans serif.

```
16 \def\ednoteshape{\sffamily}
```

The next step is to set up a counter for the editorial annotations

```
17 \newcounter{ednote}
```

`\ed@foot` The internal macro `\ed@foot` is used to actually make the annotations, it is used by the interface macros to give the annotations. It takes three arguments: A comment text that goes into the footnote, a type descriptor, and an identifiers.

```
18 \def\ed@foot#1#2#3% text, type, label
19 {\def\test{#3}\def\empty{ }\footnotetext[\value{ednote}]{%
20 {\scshape{#2}\if\test\empty\else\label{ed:#3}[{#3}]\fi:} \ednoteshape #1}}
```

`\ed@margin` The internal macro `\ed@margin` makes a `\marginpar` annotation if allowed.

```
21 \def\ed@margin#1{\ifmargins\marginpar{#1}\fi}
```

`\Ed@note` Another internal macro `\Ed@note` adds label management to `\ed@foot`

```
22 \def\Ed@note#1#2#3% text, type, label
23 {\addtocounter{ednote}{1}\message{#2!}%
24 \ifshowednotes\footnotemark[\arabic{ednote}]\ed@foot{#1}{#2}{#3}\fi}
```

`\ed@note` `\ed@note` is a variant of `\Ed@note` that also makes an identifying mark in the margin.

```
25 \def\ed@note#1#2#3% text, type, label
26 {\Ed@note{#1}{#2}{#3}\ifshowednotes\ed@margin{#2:\arabic{ednote}}\fi}
```

`\ednotelabel`

```
27 \def\ednote@label{EdNote}
28 \newcommand\ednotelabel[1]{\def\ednote@label{#1}}
```

`\ednote` with the `\Ed@note` and `\ed@note` macros it is very simple to get the desired functionality of `\Ednote` and `\ednote`:

```
29 \newcommand{\Ednote}[2] [] {\Ed@note{#2}\ednote@label{#1}}
30 \newcommand{\ednote}[2] [] {\ed@note{#2}\ednote@label{#1}}
```

`\tweaklabel`

```
31 \newcommand\tweaklabel[1] {\def\tweak@label{#1}}
32 \def\tweak@label{Tweak}
```

`\tweak` and of course for `\tweak`:

```
33 \newcommand{\tweak}[2] [] {\ed@note{#2}\tweak@label{#1}}
34 \newcommand{\Tweak}[2] [] {\Ed@note{#2}\tweak@label{#1}}
```

`\edissuelabel`

```
35 \def\edissuelabel{Issue}
36 \newcommand\edissuelabel[1] {\def\edissuelabel{#1}}
```

`\edissue`

```
37 \providecommand{\issue}[2] [] {\ed@note{#2}\edissuelabel{#1}}
38 \providecommand{\Issue}[2] [] {\Ed@note{#2}\edissuelabel{#1}}
39 \newcommand{\edissue}[2] [] {\ed@note{#2}\edissuelabel{#1}}
40 \newcommand{\edIssue}[2] [] {\Ed@note{#2}\edissuelabel{#1}}
```

`new@part` For the text status environments `newpart` and `oldpart` we also set up an internal macro that does the work.

```
41 \def\new@part#1#2#3% text, mess, start
42 {\addtocounter{ednote}{1}\edef\new@number{\theednote}\message{#2!\new@number}}
43 \ifshowednotes\ed@foot{#1}{#2}{\ed@margin{#3:\new@number}\fi}
44 \def\endnew@part#1% end
45 {\ifshowednotes\ed@margin{#1:\new@number}\fi}
```

`newpartlabels`

```
46 \def\b@newpart@label{BegNP}\def\e@newpart@label{EndNP}
47 \newcommand\newpartlabels[2] {\def\b@newpart@label{#1}\def\e@newpart@label{#2}}
```

`newpart` We instantiate it for the `newpart` environment

```
48 \newenvironment{Newpart}[1] {\new@part{#1}{New Part}\b@newpart@label} {}
49 \newenvironment{newpart}[1] {\new@part{#1}{New Part}\b@newpart@label} {\endnew@part\e@newpart@label}
```

`oldpartlabels`

```
50 \newcommand\oldpartlabels[2] {\def\b@oldpart@label{#1}\def\e@oldpart@label{#2}}
51 \def\b@oldpart@label{BegOP}\def\e@oldpart@label{EndOP}
```

`oldpart` and of course for the `oldpart` environment

```
52 \newenvironment{Oldpart}[1] %
53 {\new@part{#1}{Old Part}\b@oldpart@label\ifshowednotes\color{gray}\fi} {}
54 \newenvironment{oldpart}[1] %
55 {\new@part{#1}{Old Part}\b@oldpart@label\ifshowednotes\color{gray}\fi}
56 {\endnew@part\e@oldpart@label}
```

`todolabel`

```
57 \newcommand\todolabel[1] {\def\todo@label{#1}}
58 \def\todo@label{ToDo}
```

`todo` How we define the `todo` environment depends on the `\showednotes` switch (or the package option). If we hide annotations, `todo` is set to `comment` from the `comment` package, otherwise the body is set in sans serif font for emphasis.

```
59 \newcommand{\Todo}[2] []{\Ed@note{#2}\todo@label{#1}\ifshowednotes\bgroup\sffamily\else\comment\fi}
60 \def\endTodo{\ifshowednotes\egroup\else\endcomment\fi}
61 \newcommand{\todo}[2] []{\ed@note{#2}\todo@label{#1}\ifshowednotes\bgroup\sffamily\else\comment\fi}
62 \def\endtodo{\endTodo}
```

`todolist` How we define the `todolist` environment depends on the `\showednotes` switch (or the package option). If we hide annotations, `todolist` is set to `comment` from the `comment` package, otherwise it is set to an `itemize`.

```
63 \def\Todolist#1{% the comment
64 \ifshowednotes\message{todolist!}{\sffamily To Do: #1}\bgroup\sffamily\begin{compactenum}%
65 \else\comment%
66 \fi}
67 \def\endTodolist{\ifshowednotes\end{compactenum}\egroup\else\endcomment\fi}
68 \def\todolist#1{\ifshowednotes\ed@margin{\sffamily ToDo}\Todolist{#1}\fi}
69 \def\endtodolist{\endTodolist}
```

`musings`

```
70 \newenvironment{musings}{\ifshowednotes\color{blue}\fi}{}
```

3.1 Generating Statistics and Explanations

`\ednotemessage` The `\@ednotemessage` makes use of the counter `ednote` and generates a message.

```
71 \def\@ednotemessage{\ifnum\value{ednote}>0\typeout{}}%
72 \typeout{There are still \arabic{ednote} EdNotes, New/Oldparts, and Issues to resolve!}%
73 \typeout{}}\fi}
```

we output it automatically at the end of the log file.

```
74 \AtEndDocument{\@ednotemessage}
```

`\ednotemessage` The old `\ednotemessage` is now deprecated

```
75 \def\ednotemessage{\PackageWarning{ed}{The ‘\ednotemessage’ macro is obsolete, the message
76 is generated automatically now.}}
```

`\edexplanation` The `\edexplanation` macro makes use of the `todolist` environment.

```
77 \long\def\edexplanation{\todolist{we will use the ednote system to communicate}
78 \item use the {\tt{\char92ednote\char123author: some explanatory text\char125}}
79 like a footnote to say what you have done or what should still be
80 done\ednote{MiKo: this is an example of an ednote}. Ednotes are numbered and
81 marked in the margin for easy recognition.
82 \item use the {\tt{\char92issue\char123author: explanation of the
83 issue\char125}} variant of ednote for issues\issue{this is an example of
84 an issue} that still have to be discussed.
85 \item finally, the {\tt{todolist}} environment is a list environment that can be
86 used to mark up todo lists. This explanation is an example of a todo list, it
87 is inserted into the text in a different font.
88 \item the {\tt{newpart}} environment can be used to mark up changed text blocks.
89 {\tt{\char92begin\char123newpart\char125}} takes an argument that is
90 interpreted as a comment and is treated like an {\tt{\char92ednote}} comment.
91 \item the {\tt{oldpart}} environment is similar to {\tt{newpart}} but is used
92 for old parts of text copied from another document that still need to be
93 changed in a document.
94 \item putting the macro {\tt{\char92ednotemessage}} just before the
95 {\tt{\char92end\char123document\char125}} will generate a message with
96 cardinality information for the ednotes into the log file.
```

```
97 \item all of these text decorations and meta-annotations are only inserted into
98 the text, if the {\tt{show}} package option in the {\tt{\char92 usepackage}}
99 directive in the preamble of the document is set: {\tt{\char92
100 usepackage[show]\char123ed\char125}} will show the decorations, while
101 {\tt{\char92 usepackage\char123ed\char125}} will not. This is useful for
102 preparing ‘‘clean’’ version for outside consumption without loosing the
103 management metadata.
104 \endtodolist}
105 \end{package}
```

Change History

v1.0		v1.4	
General: First Version with Documentation	1	General: localization	1
v1.1		v1.5	
General: Added capitalized variants	1	General: todoclist now uses enumerate	1
v1.2		v1.6	
General: adding todo environment	1	General: deprecated issue for edissue to get	
v1.3		around name clashes.	1
General: rationalizing todo environment	1	nomargins option	1