Editorial Notes for LaTeX*

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

July 1, 2011

Abstract

This package defines a couple of editorial notes that simplify collaboration on a LaTeX 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.

Contents

---

*Version v1.8 (last revised 2012/01/29)

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 LaTeX 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

2.1 Package Options

As usual in LaTeX, 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   show `\usepackage[hide]{ed}`. If the `show` option is given, then the editorial notes are presented   hide 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 TeX'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.   nomargins

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

2.2 Annotation Macros

The main user macro of this package is `\ednote`. It is used say what you have done or what   \ednote 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 `\ednote`s in a page it may be difficult to find the referenced locations[1].   EdN:1 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.

The `\edissue` macro[2] is a variant of `\ednote` for issues that still have to be discussed among   \edissue the authors. For issues the labeling and cross-referencing in the optional argument is especially   Is:2 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   \issue is not defined by other packages. For this to work, the `ed` package should be loaded late in the preamble.

The `\tweak` macro[3] is a variant of `\ednote` for marking places where we have tweaked some-   \tweak thing (apart from the standard way we usually use).   Tw:3

The `todo` environment is an environment that can be used to mark up writing tasks. These   todo

---

[1] EDNOTE: this is an example of an ednote

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

[3] TWEAK: this is an example of an tweak

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.

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

The following is an example, generated form the the input `\begin{todolist}{an example todolist}`

ToDo `...\item ...\end{todolist}`. To Do: an example todolist

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

newpart the newpart environment can be used to mark up changed text blocks. `\begin{newpart}`
oldpart 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.

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.

2.3 Stubs and External Files

In some collaborative editing situations not all participants are willing to write LaTeX, but insist on developing their proposal parts in some other word procesing software. The ed class offers the
edstub edstub environment:

`\begin{edstub}[`⟨*exp*⟩`]{`⟨*file*⟩`}` marks the content of the environment as provisional (e.g. by turning it blue) and lists an external file (⟨*file*⟩) as the original. Here, ⟨*exp*⟩ is an optional reformulation of the default desription "The following blue text". The following stub

<p style="text-align:center; color:red; font-size:1.8em;">The following blue text is only a provisional stub</p>
<p style="text-align:center; color:red;">the Office document stub.doc contains more text
which will be merged for the final document</p>

here are some provisional ideas

is generated by

```
\begin{edstub}{stub.doc}
here are some provisional ideas
\end{edstub}
```

edstubURI For hyperlinking ⟨*file*⟩ the edstubURI can be used to set a URI: After setting `\edstubURI{`⟨*URI*⟩`}` with a non-empty URI, ⟨*file*⟩ is hyperlinked to ⟨*URI*⟩/⟨*file*⟩. If ⟨*file*⟩ is in the same directory as the current document, use `\edstubURI{.}`.

2.4 Generating Statistics and Explanations

`\ednotemessage`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.

`\edexplanation` 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".

2.5 Configuration

\ednoteshape   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`.

2.6 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.

3.1 Package Options

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.

3.2 Annotation Macros

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

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

\ed@footnote extends the `\ed@foot` to a complete footnote

```
21 \newcommand\ed@footnote[3]{\footnotemark[\arabic{ednote}]\ed@foot{#1}{#2}{#3}}
```

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

```
22 \newcommand\ed@margin[1]{\ifmargins\marginpar{#1}\fi}
```

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

```
23 \newcommand\Ed@note[3]% text, type, label
24 {\addtocounter{ednote}{1}\message{#2!}%
25 \ifshowednotes\ed@footnote{#1}{#2}{#3}\fi}
```

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

```
26 \newcommand\ed@note[4]% text, type, label, margin
27 {\Ed@note{#1}{#2}{#3}\ifshowednotes\ed@margin{#4:\arabic{ednote}}\fi}
```

\ednotelabel

```
28 \newcommand\ednote@label{EdNote}
29 \newcommand\ednote@margin{EdN}
30 \newcommand\ednotelabel[1]{\def\ednote@label{#1}}
31 \newcommand\ednotemargin[1]{\def\ednote@margin{#1}}
```

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

```
32 \newcommand{\Ednote}[2][]{\Ed@note{#2}\ednote@label{#1}}
33 \newcommand{\ednote}[2][]{\ed@note{#2}\ednote@label{#1}\ednote@margin}
```

\tweaklabel

```
34 \newcommand\tweaklabel[1]{\def\tweak@label{#1}}
35 \newcommand\tweak@label{Tweak}
36 \newcommand\tweakmargin[1]{\def\tweak@margin{#1}}
37 \newcommand\tweak@margin{Tw}
```

\tweak and of course for \tweak:

```
38 \newcommand{\tweak}[2][]{\ed@note{#2}\tweak@label{#1}\tweak@margin}
39 \newcommand{\Tweak}[2][]{\Ed@note{#2}\tweak@label{#1}}
```

\edissuelabel

```
40 \newcommand\edissue@label{Issue}
41 \newcommand\edissuelabel[1]{\def\edissue@label{#1}}
42 \newcommand\edissue@margin{Is}
43 \newcommand\edissuemargin[1]{\def\edissue@margin{#1}}
```

\edissue

```
44 \providecommand{\issue}[2][]{\ed@note{#2}\edissue@label{#1}\edissue@margin}
45 \providecommand{\Issue}[2][]{\Ed@note{#2}\edissue@label{#1}}
46 \newcommand{\edissue}[2][]{\ed@note{#2}\edissue@label{#1}\edissue@margin}
47 \newcommand{\edIssue}[2][]{\Ed@note{#2}\edissue@label{#1}}
```

Ed@part For the text status environments Newpart and Oldpart we also set up an internal macro that does the work.

```
48 \newenvironment{Ed@part}[3]% text, mess, start
49 {\addtocounter{ednote}{1}\edef\new@number{\theednote}\message{#2!\new@number}
50 \ifshowednotes\ed@foot{#1}{#2}{}\fi}
51 {}
```

ed@part and one that makes the marginpars.

```
52 \def\ed@part#1#2#3#4% text, mess, start, margin
53 {\Ed@part{#1}{#2}{#3}\ifshowednotes\ed@margin{#4:\new@number}\fi}
54 \def\ended@part#1{\endEd@part\ifshowednotes\ed@margin{#1:\new@number}\fi}
```

newpartlabels

```
55 \newcommand\b@newpart@label{BegNP}\newcommand\e@newpart@label{EndNP}
56 \newcommand\b@newpart@margin{BNP}\newcommand\e@newpart@margin{ENP}
57 \newcommand\newpartmargins[2]{\def\b@newpart@margin{#1}\def\e@newpart@margin{#2}}
```

newpart We instantiate it for the newpart environment

```
58 \newenvironment{Newpart}[1]{\Ed@part{#1}{New Part}\b@newpart@label}{\endEd@part}
59 \newenvironment{newpart}[1]{\ed@part{#1}{New Part}\b@newpart@label\b@newpart@margin}{\ended@part\e@newpart@ma
```

```
60 \newcommand\oldpartlabels[2]{\def\b@oldpart@label{#1}\def\e@oldpart@label{#2}}
61 \newcommand\oldpartmargins[2]{\def\b@oldpart@margin{#1}\def\e@oldpart@margin{#2}}
62 \newcommand\b@oldpart@label{BegOP}\newcommand\e@oldpart@label{EndOP}
63 \newcommand\b@oldpart@margin{BOP}\newcommand\e@oldpart@margin{EOP}
```

and of course for the `oldpart` environment

```
64 \newenvironment{Oldpart}[1]%
65 {\Ed@part{#1}{Old Part}\b@oldpart@label\ifshowednotes\color{gray}\fi}
66 {\endEd@part}
67 \newenvironment{oldpart}[1]%
68 {\ed@part{#1}{Old Part}\b@oldpart@label\b@oldpart@margin\ifshowednotes\color{gray}\fi}
69 {\ended@part\e@oldpart@margin}
```

```
70 \newcommand\todolabels[2]{\def\b@todo@label{#1}\def\e@todo@label{#2}}
71 \newcommand\todomargins[2]{\def\b@todo@margin{#1}\def\e@todo@margin{#2}}
72 \newcommand\b@todo@label{ToDo}\newcommand\e@todo@label{Done}
73 \newcommand\b@todo@margin{ToDo}\newcommand\e@todo@margin{Done}
```

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.

```
74 \newenvironment{Todo}[1]%
75 {\Ed@part{#1}{To Do}\b@todo@label\ifshowednotes\bgroup\ednoteshape\else\comment\fi}
76 {\endEd@part\e@todo@label\ifshowednotes\egroup\else\endcomment\fi}
77 \newenvironment{todo}[1]%
78 {\ed@part{#1}{To Do}\b@todo@label\b@todo@margin\ifshowednotes\bgroup\ednoteshape\else\comment\fi}
79 {\ended@part\e@todo@margin\ifshowednotes\egroup\else\endcomment\fi}
```

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.

```
80 \newenvironment{Todolist}[1]{% the comment
81 \ifshowednotes\message{todolist!}{{\ednoteshape To Do: #1}}\bgroup\ednoteshape\begin{compactenum}%
82 \else\comment%
83 \fi}
84 {\ifshowednotes\end{compactenum}\egroup\else\endcomment\fi}
85 \newenvironment{todolist}[1]{\ifshowednotes\ed@margin{{\ednoteshape ToDo}}\Todolist{#1}\fi}
86 {\endTodolist}
```

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

```
88 \def\ed@stubURI{}
89 \newcommand\edstuURI[1]{\gdef\ed@stubURI{#1}}
```

3.3 Stubs and External Files

```
90 \newif\ifhref\hreffalse
91 \AtBeginDocument{\@ifpackageloaded{hyperref}{\hreftrue}{\hreffalse}}
92 \newenvironment{edstub}[2][]
93 {\def\@test{#1}\begin{center}\huge\color{red}
94 \ifx\@test\@empty The following blue text \else #1 \fi is only a provisional stub\\\Large
95 the Office document
```

```
96 \ifx\ed@stubURI\@empty{#2}\else\ifhref\href{\ed@stubURI}{#2}\else{#2}\fi\fi\
97 contains more text\\which will be merged for the final document
98   \end{center}\color{blue}}
99 {}
```

3.4   Generating Statistics and Explanations

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

```
100 \newcommand\@ednotemessage{\ifnum\value{ednote}>0\typeout{}%
101 \typeout{There are still \arabic{ednote} EdNotes, New/Oldparts, and Issues to resolve!}%
102 \typeout{}\fi}
```

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

```
103 \AtEndDocument{\@ednotemessage}
```

\ednotemessage   The old \ednotemessage is now deprecated

```
104 \newcommand\ednotemessage{\PackageWarning{ed}{The '\ednotemessage' macro is obsolete, the message
105   is generated automatically now.}}
```

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

```
106 \newcommand\edexplanation{\todolist{we will use the ednote system to communicate}
107 \item use the {\tt{\char92ednote\char123author: some explanatory text\char125}}
108   like a footnote to say what you have done or what should still be
109   done\ednote{MiKo: this is an example of an ednote}. Ednotes are numbered and
110   marked in the margin for easy recognition.
111 \item use the {\tt{\char92issue\char123author: explanation of the
112     issue\char125}} variant of ednote for issues\issue{this is an example of
113   an issue} that still have to be discussed.
114 \item finally, the {\tt{todolist}} environment is a list environment that can be
115   used to mark up todo lists. This explanation is an example of a todo list, it
116   is inserted into the text in a different font.
117 \item the {\tt{newpart}} environment can be used to mark up changed text blocks.
118   {\tt{\char92begin\char123newpart\char125}} takes an argument that is
119   interpreted as a comment and is treated like an {\tt{\char92ednote}} comment.
120 \item the {\tt{oldpart}} environment is similar to {\tt{newpart}} but is used
121   for old parts of text copied from another document that still need to be
122   changed in a document.
123 \item putting the macro {\tt{\char92ednotemessage}} just before the
124   {\tt{char92end\char123document\char125}} will generate a message with
125   cardinality information for the ednotes into the log file.
126 \item all of these text decorations and meta-annotations are only inserted into
127   the text, if the {\tt{show}} package option in the {\tt{\char92 usepackage}}
128   directive in the preamble of the document is set: {\tt{\char92
129     usepackage[show]\char123ed\char125}} will show the decorations, while
130   {\tt{\char92 usepackage\char123ed\char125}} will not. This is useful for
131   preparing ''clean'' version for outside consumption without loosing the
132   management metadata.
133 \endtodolist}
134 ⟨/package⟩
```

Change History