

Recording Errata in L^AT_EX Documents*

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

Bendik Rønning Opstad
(bro.devel@gmail.com)

May 3, 2015

Abstract

This package provides a simple infrastructure for recording errata in L^AT_EX documents. This allows to maintain an updated version of the document (with all errors corrected) and automatically generate an errata document highlighting the difference to the published version.

1 Introduction

The life-cycle of a document does not end with its publication. After that, errors will be discovered, and have to be managed. The best way to do this is by marking errata in the text and generating the errata document from that.

2 Package Options

The `errata` package can be called with a variety of package options.

`show`= $\langle true/false \rangle$ (boolean, initially `false`)

Governs the general visibility of the errata markup. This option enables the options `footnote`^{P.2}, `margins`, `inlinechanges`^{P.2}, and `inlinemarks`^{P.2}.

`hide`= $\langle true/false \rangle$ (complementary option to `show`)

Will do the opposite of `show`, ensuring that the errata markup is totally invisible in the formatted document.

`margins`= $\langle true/false \rangle$ (boolean, initially `false`)

Enables decorations in the margin for each erratum. Setting `margins`=`false` will globally disable all margin notes.

*Version v1.0 (last revised 2015/04/16)

foots= $\langle true/false \rangle$ (boolean, initially **false**)

Enables footnote decorations for each erratum.

inlinemarks= $\langle true/false \rangle$ (boolean, initially **true**)

By default, each errata change is marked in the document with the correction encapsulated by paranthesis as shown in example 1. By setting **inlinemarks**=**false**, the inline marks will not be added.

inlinechanges= $\langle true/false \rangle$ (boolean, initially **true**)

The default behavior is to make the errata changes visible in the document inlined in the text. By setting **inlinechanges**=**false**, the document text will stay unchanged.

record= $\langle true/false \rangle$ (boolean, default **true**)

The errata entries will be written to a file named $\langle jobname \rangle$ -errata.tex so that they can be included in an errata document (see section 6).

hyperref= $\langle true/false \rangle$ (boolean, initially **true**)

This option is used to control whether the **hyperref** package will be used to create hyperlinks between each erratum, the margin notes and the errata list.

cleveref= $\langle true/false \rangle$ (boolean, initially **true**)

This package option will require the **cleveref** package, and changes how labels are created so that the registered errata corrections can be referenced with **cleveref** commands¹.

marginnote= $\langle true/false \rangle$ (boolean, initially **true**)

The default behavior is to use the **\marginpar** macro to create the errata reference in the margins. In some environments, such as floating environments, the **\marginpar** macro cannot not be used. By default, **\marginnote** from the **marginnote** package will be used to handle margin notes when necessary.

linenos= $\langle true/false \rangle$ (boolean, initially **false**)

This experimental package option utilizes the **lineno** package to get the linenumber on the page where the erratum was registered.

styles= $\langle style\ name \rangle$ (string, initially default)

The default styling of the errata markup in the errata list and the footnotes is very simple. This package option can be used to specify one or more predefined styles. Available styles are: **default**, **detailed**, **tablelist**.

requirecmddescription= $\langle true/false \rangle$ (boolean, initially **true**)

By setting this package option to **false**, the errata command macros (**\erratumAdd**^{→ P. 3}, **\erratumDelete**^{→ P. 3}, **\erratumReplace**^{→ P. 4} and **\erratumNote**^{→ P. 4}) are defined without the mandatory description argument, i.e. **\erratumAdd**[$\langle keys \rangle$]{ $\langle add \rangle$ }

¹See **Cleveref support**^{→ P. 21} for the code definition.

2.1 Default command arguments

The following package options can be used to specify a set of default settings to be passed to the errata commands described in section 5. The different settings are described in section 4.

The default arguments may be changed using the following package options:

defaultargs={*<command options>*} (key/values, default described below)

This option allows setting a set of the default arguments to be passed to the errata commands `\erratumAdd`, `\erratumReplace`^{→ P. 4}, `\erratumDelete`, and `\erratumNote`^{→ P. 4}.

The default values passed to all the commands are as follows²:

```
margin=true
foot=true
inlinemark=true
inlinechange=true
record=true
marginargs={}
```

defaultargsadd={*<command options>*} (key/values, initially empty)

This option allows setting a set of the default arguments to be passed to the `\erratumAdd` command. Any arguments set will override the respective argument set with `defaultargs`.

defaultargsdelete={*<command options>*} (key/values, initially empty)

This option allows setting a set of the default arguments to be passed to the `\erratumDelete` command. Any arguments set will override the respective argument set with `defaultargs`.

defaultargsreplace={*<command options>*} (key/values, initially empty)

This option allows setting a set of the default arguments to be passed to the `\erratumReplace`^{→ P. 4} command. Any arguments set will override the respective argument set with `defaultargs`.

defaultargsnote={*<command options>*} (key/values, initially empty)

This option allows setting a set of the default arguments to be passed to the `\erratumNote`^{→ P. 4} command. Any arguments set will override the respective argument set with `defaultargs`.

3 Errata commands

Marking errata changes in the L^AT_EX text is done using a few simple macros.

\erratumAdd[*<keys>*]{*<desc>*}{*<add>*}

marks up a correction by adding the text *<add>* for addition with *<desc>* describing the intended correction.

\erratumDelete[*<keys>*]{*<desc>*}{*<delete>*}

marks up a correction by deleting the text *<delete>* with *<desc>* describing the intended correction.

²The code defining the default settings are defined in `Default command arguments`^{→ P. 15}.

`\erratumReplace[⟨keys⟩]{⟨desc⟩}{⟨old⟩}{⟨new⟩}`

marks up a correction by replacing the text `⟨old⟩` with `⟨new⟩`.

`\erratumNote[⟨keys⟩]{⟨desc⟩}{⟨note⟩}`

creates a note entry that cannot be described by the other commands.

3.1 Errata example

Example 1 shows all editing operations in action.

Example

```
Here we have \erratumAdd{forgotten word}{three} errata in one \erratumDelete{
superfluous phrase}{darned} long \erratumReplace{translated}{Zeile}{line}.
```

Example 1: Some short errata

All of these macros mark the location of the errata in the margin and document the changes in footnote-like structures. The text in example 1 is rendered as follows:

Here we have [three]_a¹ errata in one []_a² long [line]_r³

Err(1)
Err(2)
Err(3)

If the package option `hyperref`^{→P. 2} is enabled, the margin note will link to the entry in the errata list produced by calling `\PrintErrata`^{→P. 8}.

3.2 Errata environment

```
\begin{erratum}[⟨keys⟩]{⟨desc⟩}
  ⟨environment content⟩
\end{erratum}
```

Extended errata group multiple editing operations into a coherent group via the `erratum` environment. The first argument of this environment is an explanation as for the short errata above. The `erratum` environment provides local versions of the editing markup macros, which behave like those, but lack the first (explanation) argument, which is already given in the environment that contains them.

Example

```
\begin{erratum}[date=2006-07-19,reported-by=Michael Kohlhase]{old should be new}
This is a test of a long erratum
\begin{enumerate}
  \item We can replace \eReplace{oldtext}{newtext}
  \item and \eAdd{new text}
  \item and finally delete old text\eDelete{alltogether}
\end{enumerate}
\end{erratum}
```

¹ERRATUM! forgotten word (added text)

²ERRATUM! superfluous phrase (deleted “darned”)

³ERRATUM! translated (original text was: “Zeile”)

Example 2: An extended erratum with local correction markers

BErr(4)

The text in example 2 would be rendered as

this is a test of a long erratum

1. We can replace [newtext]_r^{4:1}
2. and [new text]_a
3. and finally delete old text []_d^{4:2}

EErr(4)

The `erratum` environment should also be used in situations where the error occurs in an environment, where normal $\text{T}_\text{E}\text{X}$ / $\text{L}^\text{A}\text{T}_\text{E}\text{X}$ processing is suspended, e.g. a `verbatim` environment. In this case, we can use it to attach correction information via the environment, but do not use the local change documentations.

4 Command Options

Each of the errata commands take an optional list of comma separated key/values as argument. This can be used to specify per-command options for each errata entry.

`margin`= $\langle true/false \rangle$ (boolean, defaults to `true`)

Allows to override the default `margin` value for this erratum.

`foot`= $\langle true/false \rangle$ (boolean, defaults to `true`)

Allows to override the default `foot` value for this erratum.

`inlinechange`= $\langle true/false \rangle$ (boolean, defaults to `true`)

Allows to override the default `inlinechange` value for this erratum.

`inlinemark`= $\langle true/false \rangle$ (boolean, defaults to `true`)

Allows to override the default `inlinemark` value for this erratum.

`label`= $\langle name\ of\ label \rangle$ (initially empty)

Allows specifying a custom label name to be registered for this errata entry.

`uselabel`= $\langle name\ of\ label \rangle$ (string, defaults to `errata<X>`)

Allows specifying a label to use instead of the default label generated for each registered erratum entry (`errata<X>` where X is the erratum number). This is necessary in environments where only one label may exist, such as in the `equation` environment.

`marginargs`= $\{ \langle key/valoptions \rangle \}$ (list, initially empty)

Allows specifying custom arguments to be passed to the marginnote formatter.

`usemarginnote`[$= \langle length \rangle$] (length, initially `0pt`)

Forces the use of `\marginnote` for the errata command. An optional value can be specified which denotes how much to vertically shift the margin note.

⁴ERRATUM! OLD SHOULD BE NEW

4:1 was: oldtext

4:2 deleted: altogether

5 Errata metadata

5.1 Data keys

For each errata command, a set of meta-data can be specified for an errata entry. The following keys can be used to registered additional data:

date=*<date when registered>* (string, initially empty)

Allows specifying the date when the erratum was registered.

reported-by=*<name of reporter>* (string, initially empty)

Allows specifying the name of the reporter.

description=*<description>* (string, initially empty)

Allows specifying the description of the change.

If the package option `requirecmddescription`^{P.2} is true (the default), This key defaults to the value specified in the *<desc>* field.

location=*<location hint>* (string, initially empty)

Used to specify the location in the text where the change is made.

type=*<type of correction>* (string, initially empty)

Used to specify what type of change is registered, e.g. grammar, clarification, layout.

Remember to encapsulate any of values in curly braces if it contains a comma.

5.2 Formatter keys

For each of the metadata keys specified in section 5.1, a formatter is defined for the errata list and the footnotes. The formatter defines a custom formatting rule which will be applied to the input value before being used.

Additionally, the following formatter keys exist which may **not** be passed as argument to the errata commands:

pageref (name of label, errata<X>)

Formats a reference to the page number where the erratum is located.

actiondescription

the formatted description of the action

action

The action name

erratumlistitem

This key is used to insert an `\item[]` in the errata list.

erratumlistitemtitle

The title for the erratum in the errata list

erratumlisttarget

The name used when creating a `\label` target in the errata list.

marginfmt/marginnote

Formats the marginnote

5.3 Setting the format for footnotes and errata list

The formatting of the footnotes and errata list can be changed using the two macros `\errataSetListFormat` and `\errataSetFootFormat`.

The argument $\langle keys \rangle$ should specify a comma separated list of keys as specified in section 5³.

\errataSetFootFormat $\{\langle keys \rangle\}$

Allows changing how the footnotes are formatted.

```
\errataSetFootFormat{+description, +actiondescription}
```

\errataSetListFormat $\{\langle keys \rangle\}$

The default format for the errata list is defined with:

```
\errataSetListFormat{+erratumlisttarget, +description}
```

5.4 Formatting the keys

The format of each of the keys can be set using the macro `\errataSetKeyFormat`.

\errataSetKeyFormat $\{\langle key \rangle\}\{\langle format \rangle\}$

should be used to set the formatter key for a meta data key.

The default format for the keys are as follows⁴:

³See `Errata default style definition` → P. 17 for the code definition

⁴This code is defined at `Default formatters` → P. 18.

```

%%
\NewDocumentCommand{\erratalistitem}{ m }{\item[#1]}
\errataset{%
  styles/basekeyformats/.style={%
    actionidformat/add/.setStrKeyFmt={##1}{Addition},
    actionidformat/delete/.setStrKeyFmt={##1}{Deletion},
    actionidformat/note/.setStrKeyFmt={##1}{Note},
    actionidformat/replace/.setStrKeyFmt={##1}{Correction},
    %%
    listformats/listbeforenewgroup/.setStrKeyFmt={##1}{},
    listformats/listafternewgroup/.setStrKeyFmt={##1}{},
    listformats/listbeforesamegroup/.setStrKeyFmt={##1}{},
    listformats/listaftersamegroup/.setStrKeyFmt={##1}{},
    %% The name of the errata labels
    errformats/errfmt/.setStrKeyFmt={##1}{Err(##1)},
    errformats/eerrbeginfmt/.setStrKeyFmt={##1}{BErr(##1)},
    errformats/eerrendfmt/.setStrKeyFmt={##1}{EErr(##1)},
    %%
    listformats/erratumlabel/.setStrKeyFmt={##1}{\@errata@labelname},
    listformats/erratumlistitem/.setStrKeyFmt={##1}{\erratalistitem{##1}},
    listformats/erratumlisttarget/.setStrKeyFmt={##1}{\@errata@htargeterratum{##1}{}},
    listformats/erratumnamedlink/.setStrKeyFmt={##1+##2}{%
      \@errata@hlink{##1}{\protect\makebox[\@errata@groupindent][l]{##2}}%
    },
    %%
    listformats/erratalistbegin/.setStrKeyFmt={##1}{\makeatletter\begin{description}},
    listformats/erratalistend/.setStrKeyFmt={##1}{\end{description}\makeatother},
  },
}
\errataset{%
  defaultkeyformats/.style={%
    %% Formats for the footnotes
    footformats/description/.setStrKeyFmt={##1}{~##1},
    footformats/action/.setStrKeyFmt={##1}{~##1},
    footformats/actiondescription/.setStrKeyFmt={##1}{~##1},
    %% Set the format for the action description for each erratum command
    actionformat/add/.setStrKeyFmt={##1}{(added text)},
    actionformat/delete/.setStrKeyFmt={##1}{(deleted ‘‘##1’’)},
    actionformat/replace/.setStrKeyFmt={##1+##2}{(original text was: ‘‘##2’’)},
    %% Errata list format
    listformats/description/.setStrKeyFmt={##1}{~##1},
    listformats/erratumlistitemtitle/.setStrKeyFmt={##1+##2}{%
      \@errata@hlink{##1}{\protect\makebox[\@errata@groupindent][l]{##2}}},
    listformats/type/.setStrKeyFmt={##1}{Type: ##1 },
    listformats/date/.setStrKeyFmt={##1}{(Date reported: ##1)},
    listformats/reported-by/.setStrKeyFmt={##1}{Reported by: ##1},
    listformats/action/.setStrKeyFmt={##1}{~##1},
    listformats/actiondescription/.setStrKeyFmt={##1}{~##1},
  },
}

```

6 Generating Statistics and Errata Documents

\PrintErrata

This command inserts the errata list at the current position.

`\printerrata[⟨postfix(-errata.tex)⟩]{⟨prefix⟩}`

This command allows to print the errata for another document. This command is useful when generating errata documents for published works. Say we have a book with a driver file `thebook.tex`, into which we have incorporated errata markup using the infrastructure detailed above. Then we have a new document called e.g. `theerrata.tex` which has the form given in example 3. Note that we have used `\printerrata{thebook}` to include the errata notices generated from `thebook.tex`.

```
\begin{errata}
  environment content
\end{errata}
```

The errata list produced when the `record`^{P.2} option is enabled is placed inside this environment.

```
\erratumItem[macro]{title}{text}
```

Errata can be marked up using the `\erratumItem` macro in the `errata` environment. `\erratumItem` takes one optional, and two required arguments.

The first argument (`[macro]`) can be used to specify a macro containing a value that will be expanded after the other values, i.e. at the end.

The second argument `{title}` will be formatted using the `listformats/erratumlistitem`^{P.6} key formatter. For the default formatting of the errata list this is used to insert a `\item[title]` before the remaining errata meta data specified in the third argument `{text}`

Normally it is not necessary to manually use this macro, as the `record`^{P.2} option will write an `erratumItem` to the errata for each registered erratum.

```
\erratamessage
```

Putting the macro `\erratamessage` just before the `\end{document}` will generate a message with cardinality information for the errata into the log file.

In the errata document in example 3 we postulate that we keep an updated version of The Book online⁵ using the infrastructure provide by the `errata` package. In the updated version of `thebook.tex`, it can be useful to tabulate the errata as well, e.g. in a section in the appendix. This can be done by the `\PrintErrata`^{P.8} command. Note that this command needs to close the errata file `thebook-errata.tex` therefore we need a `\newpage` to clear the queue of waiting `\writes` before `thebook-errata.tex` can be loaded (otherwise we may be missing the errata from the last page).

⁵And indeed it is good practice to do so if the copyright agreement with the publisher allows this.

Example

```

\documentclass{article}
\usepackage[hide]{errata}
\title{Errata for The Book}
\begin{document}
\maketitle
\begin{abstract} This document tracks the errata in The Book. \end{abstract}
\section{Introduction}
The errata of The Book are tracked in this document, whose newest version
can be found at \url{../berrata.pdf}. A version of The Book that contains
all errata corrections (and markup of what changed) can be found at
\url{../book.pdf}.

In the following we will tabulate the errata in document order. Their
location will be referenced by the section they appear in rather than the
page number, since we do not expect the former to change in the errata
correction process.

\section{The Errata in The Book}
\printerrata{thebook}
\end{document}

```

Example 3: A Sample Errata Document

7 Limitations

If you want to contribute, please contact the author or make a pull request at <https://github.com/KWARC/LaTeX-errata>

7.1 Support for special environments

When registering an erratum in environments such as `equation` (from `amsmath` package), labels cannot be created freely, therefore the label already registered in the environment must be passed to the errata command.

Example 4 shows example where the label is specified manually.

Title

```
\begin{equation}\label{customlabel}
\begin{split}
a=a \quad \backslash\backslash
b=b \quad \backslash erratumAdd[uselabel=customlabel]{label}{custom label example}
\end{split}
\end{equation}
```

$$a = a$$

$$b = b[customlabel example]_a^5$$

$$(1)$$

⁵ERRATUM! label (added text)
⁵ERRATUM! label (added text)

Example 4: Some short errata

7.2 Floating environments

When registering errata corrections inside floating environments, some manual adjustments may be required. Example 5 shows how the margins can be adjusted manually.

Title	
<p>In this sentence<code>\erratumAdd[inline mark=false,label={comma1}]{added comma}{,}</code> we added a comma with inline mark disabled<code>\erratumAdd[inline mark=true,label={comma2}]{added another comma}{,}</code> as well as a comma with inlinemarking.</p> <p>Inside environments where <code>\refAuxcs{marginnote}</code> must be used, it may be necessary to manually adjust the vertical placement of the margin note when multiple notes overlap.</p> <p>As this code is processed inside a <code>\refAux{tcolorbox}</code>, <code>\refAuxcs{marginnote}</code> must be used instead of <code>\refAuxcs{marginpar}</code>. As we can see, <code>\Cref{comma1}</code> and <code>\cref{comma2}</code> are overlapping in the margin as <code>\refAuxcs{marginnote}</code> does not handle stacking multiple notes. Therefore, we must manually ask <code>\refAuxcs{marginnote}</code> to shift the second margin value, such as with <code>\erratumAdd[inline mark=true]{added missing word}{these}</code> two <code>\erratumAdd[inline mark=true,usemarginnote=20pt]{added another missing word}{corrections}</code>.</p>	
<p>In this sentence, we added a comma with inline mark disabled⁶,⁷_a as well as a comma with inlinemarking.</p> <p>Inside environments where <code>\marginnote</code> must be used, it may be necessary to manually adjust the vertical placement of the margin note when multiple notes overlap.</p> <p>As this code is processed inside a <code>tcolorbox</code>, <code>\marginnote</code> must be used instead of <code>\marginpar</code>. As we can see, Err(6) and Err(7) are overlapping in the margin as <code>\marginnote</code> does not handle stacking multiple notes. Therefore, we must manually ask <code>\marginnote</code> to shift the second margin value, such as with ⁸[these]_a⁹ two [corrections]_a.</p>	<p>Err(6)</p> <p>Err(8)</p>
<p>⁶ERRATUM! added comma (added text) ⁷ERRATUM! added another comma (added text) ⁸ERRATUM! added missing word (added text) ⁹ERRATUM! added another missing word (added text)</p>	<p>Err(9)</p>

Example 5: Example showing how to adjust the margin notes

8 The Implementation

```

\NeedsTeXFormat{LaTeX2e}[1999/12/01]
\ProvidesPackage{errata}[2015/04/16 v1.0 Errata Management]
\RequirePackage{xifthen}
\RequirePackage{pgfkeys}
\RequirePackage{etoolbox}
\RequirePackage{xpatch}
\RequirePackage{xparse}
\RequirePackage{xinttools} % defines \xintFor macro
\RequirePackage[family=@errata@]{strkeyformatter}
\RequirePackage{kvoptions}[2009/07/21]

```

Setup package options

```

\SetupKeyvalOptions {
  family = Errata,
  prefix = @errata0pt@,
}

```

show The show package option shows everything

```
\DeclareBoolOption{show}
```

hide The hide package option hides everything

```
\DeclareComplementaryOption{hide}{show}
```

margins The margins package option

```
\DeclareBoolOption{margins}
```

foots The foots package option

```
\DeclareBoolOption{foots}
```

record The record package option shows everything

```
\DeclareBoolOption[true]{record}
```

requirecmddescription The requirecmddescription package option

```
\DeclareBoolOption[true]{requirecmddescription}
```

hyperref The hyperref package option

```
\DeclareBoolOption[true]{hyperref}
```

cleveref The cleveref package option

```
\DeclareBoolOption[true]{cleveref}
```

marginnote The marginnote package option

```
\DeclareBoolOption[true]{marginnote}
```

linenos The linenos package option

```
\DeclareBoolOption[false]{linenos}
```

<code>environmenttestmode</code>	<p>The <code>environmenttestmode</code> package option allows to set when <code>\marginnote</code> is used instead of <code>\marginpar</code>. Errata will try to detect when <code>\marginnote</code> should be used instead of <code>\marginpar</code>, and this option allows for finetuning how safe the test is. The default (<code>environmenttestmode=0</code>) will only use <code>\marginnote</code> when it knows for certain that it cannot use <code>\marginpar</code>. This is in <code>math</code> mode, in <code>inner</code> mode, and in <code>tcolorbox</code>'s <code>tcb@savebox</code> environment. There are certainly many more that should be added to this list. By setting <code>environmenttestmode=1</code>, <code>\marginnote</code> will be used whenever <code>\@capttype</code> is defined, which should include most floating environments such as <code>table</code> and <code>figure</code>. The strictest mode <code>environmenttestmode=2</code> will force <code>\marginnote</code> to be used whenever the current environment is not 'document', which means any environment outside of the root document.</p> <pre>\DeclareStringOption[0]{environmenttestmode}</pre>
<code>uniquelistlabels</code>	<p>The <code>uniquelistlabels</code> package option Disabled by default (0), if option set, default level is 3.</p> <pre>\DeclareStringOption[0]{uniquelistlabels}[3]</pre>
<code>inlinemarks</code>	<p>The <code>inlinemarks</code> package option</p> <pre>\DeclareBoolOption[true]{inlinemarks}</pre>
<code>inlinechanges</code>	<p>The <code>inlinechanges</code> package option</p> <pre>\DeclareBoolOption[true]{inlinechanges}</pre>
<code>styles</code>	<p>The <code>styles</code> package option</p> <pre>\DeclareStringOption[default]{styles}</pre>
<code>defaultargs</code>	<p>The <code>defaultargs</code> package option</p> <pre>\DeclareStringOption[]{defaultargs}</pre>
<code>defaultargsadd</code>	<p>The <code>defaultargsadd</code> package option</p> <pre>\DeclareStringOption[]{defaultargsadd}</pre>
<code>defaultargsdelete</code>	<p>The <code>defaultargsdelete</code> package option</p> <pre>\DeclareStringOption[]{defaultargsdelete}</pre>
<code>defaultargsreplace</code>	<p>The <code>defaultargsreplace</code> package option</p> <pre>\DeclareStringOption[]{defaultargsreplace}</pre>
<code>defaultargsnote</code>	<p>The <code>defaultargsnote</code> package option</p> <pre>\DeclareStringOption[]{defaultargsnote}</pre>

```

\ProcessKeyvalOptions{Errata}
%% Utility macros
\input{erratapgfkeysextra}
\NewDocumentCommand{\@errata@requirepackage}{ m }{%
  \@ifpackageloaded{#1}{%
    \PackageWarning{errata}{The current errata settings require the #1 package to be loaded.
      Loading package #1...}
    \RequirePackage{#1}
  }
}
\def\errataset{\pgfkeys{/@errata@}}
\DeclareExpandableDocumentCommand{\@errata@setvalue}{ m } {\pgfkeys{/@errata@/#1}}
\DeclareExpandableDocumentCommand{\@errata@setvalues}{ 0{ } m } {\pgfkeys{/@errata@/#1.cd, #2}}
\DeclareExpandableDocumentCommand{\@errata@getvalue}{ m } {\pgfkeysvalueof{/@errata@/#1}}
\NewDocumentCommand{\@errata@isnotempty}{ m m } {\ifthenelse{\equal{#1}{}}{#2}}
\NewDocumentCommand{\@errata@isempty}{ m m } {\ifthenelse{\equal{#1}{}}{#2}}
\NewDocumentCommand{\@errata@ifstrequal}{ m m m } {\ifthenelse{\equal{#1}{#2}}{#3}}
%%

```

Default command arguments

Define the custom default arguments for the errata command

```

\pgfkeys{
  /@errata/cmdkeys/.is family, /@errata/cmdkeys,
  defaultargs/.style={record=true, foot=true, margin=true, marginargs={},
    lineno=true, inlinemark=true, inlinechange=true},
  defaultargsadd/.style={},
  defaultargsreplace/.style={},
  defaultargsdelete/.style={},
  defaultargsnote/.style={},
  /handlers/.errataapplystyles/.code={\xintFor ##9 in {#1} \do {\errataset{styles/##9}}},
}
%% Add any default arguments specified by the user
\@errata@setvalues[cmdkeys/]{defaultargs/.append style/.expand once={\@errataOpt@defaultargs}}
\@errata@setvalues[cmdkeys/]{defaultargsadd/.append style/.expand once={\@errataOpt@defaultargsadd}}
\@errata@setvalues[cmdkeys/]{defaultargsreplace/.append style/.expand once={\@errataOpt@defaultargsreplace}}
\@errata@setvalues[cmdkeys/]{defaultargsdelete/.append style/.expand once={\@errataOpt@defaultargsdelete}}
\@errata@setvalues[cmdkeys/]{defaultargsnote/.append style/.expand once={\@errataOpt@defaultargsnote}}
%%
\AtBeginDocument{%
  \ifboolexpr{ bool{\@errataOpt@linenos} and not test{\@ifpackageloaded{lineno}} } {%
    \@errata@requirepackage{lineno}
  }{}
}
%%
\if@errataOpt@show
\@errataOpt@marginstrue%
\@errataOpt@footstrue%
\@errataOpt@inlinemarkstrue%
\@errataOpt@inlinechangestruetrue%
\fi
%%
\@ifpackageloaded{hyperref}{\@errataOpt@hyperreffalse }

```

```

\ifpackageloaded{cleveref}{\@errataOpt@clevereffalse }
\iferrataOpt@marginnote
\ifpackageloaded{marginnote}{\RequirePackage{marginnote}}
\fi
%%

\newcounter{errata@erratum}
\newcounter{erratum@note}[errata@erratum]
\iferrataOpt@record\newwrite\errata@file
\immediate\openout\errata@file=\jobname-errata.tex
\AtEndDocument{\closeout\errata@file}\fi
%%

\newlength{\errata@groupindent}
\setlength\errata@groupindent{14.7pt}
\newcounter{errata@uniquelistlabels}
\setcounter{errata@uniquelistlabels}{\errataOpt@uniquelistlabels}
\newcounter{errata@environmenttestmode}
\setcounter{errata@environmenttestmode}{\errataOpt@environmenttestmode}

\NewDocumentCommand{\errata@declareKey}{ O{UNDEF} m }
{\errata@setvalue{#2/.initial={#1}}}%
%%

\NewDocumentCommand{\errata@declareStoreKey}{ O{UNDEF} m }
{%% Basically 'estore in', but must build macro name from argument
\pgfkeys{/errata@/#2/.code={\expandafter\edef\csname @errata@#2\endcsname{##1}}}%
\pgfkeys{/errata@/#2={#1}}%
}
%%

\NewDocumentCommand{\errata@setcmdkeys}{ O{0} m }{%
\xintFor ##9 in {#2} \do {%
\edef\errata@key{##9}%
\errata@isnotempty{##9} {%
\errata@setvalue{cmdkeys/\errata@key/.initial=#1}%
}}
}
%%

\NewDocumentCommand{\errataSetFootFormat}{ m }{%
\errata@declareStoreKey[#1]{footformat}%
}

\NewDocumentCommand{\errataSetListFormat}{ m }{%
\errata@declareStoreKey[#1]{listformat}%
}

\errata@declareStoreKey[type, description, location, date, id, pageref, lineref,
inlinemark, inlinechange, uselabel, label, margin, foot, record, lineno,
marginargs, usemarginnote, reported-by, actionid, action, actionargs, actiondescription,
erratumlistitem, erratumlistitemtitle, erratumlisttarget,
listbeforenewgroup, listafternewgroup, listbeforeamegroup, listaftersamegroup,
erratumlabel, erratumname, erratumnamedlink, customvalue]{formatkeys}% Defines \@errata@formatkeys
%%

\errata@setcmdkeys{\errata@formatkeys}
\errata@setvalue{cmdkeys/usemarginnote/.default={0pt}}
%%

```

`\errataSetKeyFormat` Macro for setting the format of a key

```

\NewDocumentCommand{\errataSetKeyFormat}{ O{ } m m }{% PGF argument pattern, key, formatted output
\StrFmtSetKeyFormat[#1]{#2}{#3}%

```