The **atenddvi** package

Heiko Oberdiek[*]

2020-11-11 v1.5

This package is unneeded and does nothing with a LaTeX version 2020-10-01 or newer as `\AtBeginDvi` is then provided by the format.

Abstract

LaTeX offers `\AtBeginDvi`. This package **atenddvi** provides the counterpart `\AtEndDvi`. The execution of its argument is delayed to the end of the document at the end of the last page. Thus `\special` and `\write` remain effective, because they are put into the last page. This is the main difference to `\AtEndDocument`.

Contents

1 Documentation

`\AtEndDvi` {⟨*code*⟩}

Macro `\AtEndDvi` provides a hook mechanism to put ⟨*code*⟩ at the end of the last output page. It is the logical counterpart to `\AtBeginDvi`. Despite the name the output type DVI, PDF or whatever does not matter.

---

[*] Please report any issues at https://github.com/ho-tex/atenddvi/issues

Unlike `\AtBeginDvi` the ⟨*code*⟩ is not put in a box and therefore executed immediately. The hook for `\AtEndDvi` is based on a macro similar to `\AtBeginDocument` or `\AtEndDocument`. The execution of ⟨*code*⟩ is delayed until the hook is executed on the last page.

Commands such as `\special` or `\write` (not the `\immediate` variant) must go as nodes into the contents of a page to have the desired effect. When the hook for `\AtEndDocument` is executed, the last intended page may already be shipped out. Therefore `\special` or `\write` cannot be used in a reliable way without generating new page.

This gap is closed by `\AtEndDvi` of this package `atenddvi`. If the document is compiled the first time, the package remembers the last page in a reference. In the sceond run, it puts the hook on the page that has been detected in the previous run as last page. The package detectes if the number of pages has changed, and then generates a warning to rerun LaTeX.

2 Implementation

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{atenddvi}%
  [2020-11-11 v1.5 At end DVI hook (HO)]%
\providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
\IfFormatAtLeastTF{2020/10/01}
  {%
   \PackageWarning{atenddvi}
     {Functionality of this package is already\MessageBreak
      provided by LaTeX.\MessageBreak\MessageBreak
      It is therefore no longer necessary to load it\MessageBreak
      and you can safely remove it.\MessageBreak
      Found on}{}%
  }{\input{atenddvi-2019-12-11.sty}}
\IfFormatAtLeastTF{2020/10/01}{}{\endinput}

⟨/package⟩
```

```
⟨*packagefrozen⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{atenddvi}%
  [2020-11-11 v1.5 At end DVI hook / legacy frozen code (HO) ]%
```

Load the required packages

```
\RequirePackage{zref-abspage,zref-lastpage}[2007/03/19]
\RequirePackage{atbegshi}
```

`\AtEndDvi@Hook`

Macro `\AtEndDvi@Hook` is the data storage macro for the code that is executed later at end of the last page.

```
\let\AtEndDvi@Hook\@empty
```

`\AtEndDvi`

Macro `\AtEndDvi` is called in the same way as `\AtBeginDocument`. The argument is added to the hook macro.

```
\newcommand*{\AtEndDvi}{%
  \g@addto@macro\AtEndDvi@Hook
}
```

`\AtEndDvi@AtBeginShipout`

```
\def\AtEndDvi@AtBeginShipout{%
  \begingroup
```

The reference 'LastPage' is marked used. If the reference is not yet defined, then the user gets the warning because of the undefined reference and the rerun warning at the end of the compile run. However, we do not need a warning each page, the first page is enough.

```
    \ifnum\value{abspage}=1 %
      \zref@refused{LastPage}%
    \fi
```

The current absolute page number is compared with the absolute page number of the reference 'LastPage'.

```
    \ifnum\zref@extractdefault{LastPage}{abspage}{0}=\value{abspage}%
```

`\AtEndDvi@LastPage`

We found the right page and remember it in a macro.

```
      \xdef\AtEndDvi@LastPage{\number\value{abspage}}%
```

The hook of `\AtEndDvi` is now put on the last page after the contents of the page.

```
      \global\setbox\AtBeginShipoutBox=\vbox{%
        \hbox{%
          \box\AtBeginShipoutBox
          \setbox\AtBeginShipoutBox=\hbox{%
            \begingroup
              \AtEndDvi@Hook
            \endgroup
          }%
          \wd\AtBeginShipoutBox=\z@
          \ht\AtBeginShipoutBox=\z@
          \dp\AtBeginShipoutBox=\z@
          \box\AtBeginShipoutBox
        }%
      }%
```

We do not need the every page hook.

```
      \global\let\AtEndDvi@AtBeginShipout\@empty
```

The hook is consumed, `\AtEndDvi` does not have an effect.

```
      \global\let\AtEndDvi\@gobble
```

Make a protocol entry, which page is used by this package as last page.

```
      \let\on@line\@empty
      \PackageInfo{atenddvi}{Last page = \AtEndDvi@LastPage}%
    \fi
  \endgroup
}
```

`\AtEndDvi@AtBeginDocument`

In order to get as late as possible in the chain of the every shipout hook, the call of `\AtBeginShipout` is delayed.

```
\def\AtEndDvi@AtBeginDocument{%
  \AtBeginShipout{\AtEndDvi@AtBeginShipout}%
```

`\AtEndDvi@Check`

After `\AtEndDocument` LaTeX reads its `.aux` files again. Code in `\AtEndDocument` could generate additional pages. This is unlikely by code in the `.aux` file, thus we use the `.aux` file to run macro `\AtEndDvi@Check` for checking the last page.

During the first reading of the `.aux` file, `\AtEndDvi@Check` is disabled, its real meaning is assigned afterwards.

```
  \if@filesw
    \immediate\write\@mainaux{%
      \string\providecommand\string\AtEndDvi@Check{}%
    }%
    \immediate\write\@mainaux{%
      \string\AtEndDvi@Check
    }%
  \fi
  \let\AtEndDvi@Check\AtEndDvi@CheckImpl
}
```

```
\AtBeginDocument{\AtEndDvi@AtBeginDocument}
```

`\AtEndDvi@CheckImpl` First check is whether a last page was found at all. Secondly the found last page is compared with the real last page.

```
\def\AtEndDvi@CheckImpl{%
  \@ifundefined{AtEndDvi@LastPage}{%
    \PackageWarningNoLine{atenddvi}{%
      Rerun LaTeX, last page not yet found%
    }%
  }{%
    \ifnum\AtEndDvi@LastPage=\value{abspage}%
    \else
      \PackageWarningNoLine{atenddvi}{%
        Rerun LaTeX, last page has changed%
      }%
    \fi
  }%
}
```

```
⟨/packagefrozen⟩
```

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/atenddvi/atenddvi.dtx The source file.

CTAN:macros/latex/contrib/atenddvi/atenddvi.pdf Documentation.

3.2 Package installation

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain TeX:

```
tex atenddvi.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
atenddvi.sty            → tex/latex/atenddvi/atenddvi.sty
atenddvi-2019-12-11.sty → atenddvi-2019-12-11.sty
atenddvi.pdf            → doc/latex/atenddvi/atenddvi.pdf
atenddvi.dtx            → source/latex/atenddvi/atenddvi.dtx
```

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

---

[1] CTAN:pkg/atenddvi

3.3 Refresh file name databases

If your TeX distribution (TeX Live, MiKTeX, . . .) relies on file name databases, you must refresh these. For example, TeX Live users run `texhash` or `mktexlsr`.

3.4 Some details for the interested

Unpacking with LaTeX. The `.dtx` chooses its action depending on the format:

plain TeX: Run docstrip and extract the files.

LaTeX: Generate the documentation.

If you insist on using LaTeX for docstrip (really, docstrip does not need LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{atenddvi.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLaTeX:

```
pdflatex atenddvi.dtx
makeindex -s gind.ist atenddvi.idx
pdflatex atenddvi.dtx
makeindex -s gind.ist atenddvi.idx
pdflatex atenddvi.dtx
```

4 History

[2007/03/20 v1.0]

- First version.

[2007/04/17 v1.1]

- Package atbegshi replaces package everyshi.

[2016/05/16 v1.2]

- Documentation updates.

[2019/12/05 v1.3]

- Documentation updates.

[2019-12-11 v1.4]

- Updated

[2020-11-11 v1.5]

- The package is now unneeded as LaTeX formats newer than 2020-10-01 provide `\AtBeginDvi` and so does nothing in this case. The old code has been frozen.

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.