Generating multiple versions of a document for different audiences from the same source*†

Boris Veytsman ‡

2021/10/02, v1.04

Abstract

The *multiaudience* package allows to generate several versions of the same document for different audiences

Contents

1 Introduction **2**

2 User's guide **2**
2.1 Invocation 2
2.2 Visibility scopes 3
2.3 Commands and environments 4
2.4 Verbatim text 4
2.5 Extensions 5
2.6 Combined audiences 5

3 Implementation **6**
3.1 Switches and defaults 6
3.2 Key-value interface 6
3.3 Main command 7
3.4 Main environment 7
3.5 Extensions 7

†This package was commissioned by Neadwerx, http://www.neadwerx.com/
‡borisv@lk.net

1 Introduction

Suppose you want to present the same paper, say, software design document, to software developers and executives. You expect that your intended readers have different interests and expectations: some paragraphs will be interesting only to the developers, while some—only to the executives. On the other hand, some paragraphs will be interesting to both. Basically you want two different versions of the document, one for each audience. Another example may be the situation where you need a public and a confidential versions of the same document (or even classified and unclassified ones).

Of course standard version controls tools like *subversion*, *git* or *CVS* allow you to create branches of your document, but in many cases this is not a good solution. If the versions have many paragraphs in common, you will find yourself constantly merging and branching your revisions.

The package *multiaudience* offers an alternative solution to the problem. You create a LaTeX document, clearly marking some parts for different audiences. When you process the document, you indicate, for which audience the current output is intended. The corresponding parts will be chosen.

2 User's guide

2.1 Invocation

`\CurrentAudience`
`\DefCurrentAudience`

To use the package you need first to define the macro `\CurrentAudience`. There are two ways to do it: a LaTeX-ish macro `\DefCurrentAudience{`⟨*audience*⟩`}`

```
\DefCurrentAudience{execs}
```

or TeX-ish `\def` primitive:

```
\def\CurrentAudience{execs}
```

You may set this parameter outside the document itself like this:

```
pdflatex "\def\CurrentAudience{execs}\input{file}"
```

This trick allows one to generate all versions of output from the command line:

```
pdflatex -jobname file-execs "\def\CurrentAudience{execs}\input{file}"
pdflatex -jobname file-devs "\def\CurrentAudience{devs}\input{file}"
...
```

Then put in the preamble of your document *after* defining the current audience

```
\usepackage{multiaudience}
```

`\SetNewAudience` After calling the package you need to list all possible audience using the command `\SetNewAudience{`⟨*audience*⟩`}`, for example,

```
\SetNewAudience{execs}
\SetNewAudience{devs}
\SetNewAudience{admins}
```

Now you can mark the portions of your document, for example,

```
\begin{shownto}{execs, admins}
  Text for execs and admins
\end{shownto}
\begin{shownto}{-, execs}
  Text for everybody but execs
\end{shownto}
```

2.2 Visibility scopes

The package contains a number of commands and environments, that mark certain portions of document for different audiences. They all have as an argument an `{`⟨*audiences*⟩`}` list. It is a comma-separated list of audiences (defined by the command `\SetNewAudience`, see Section 2.1), optionally starting with the keyword "-". This argument defines the visibility scope of the command or environment.

The rules are:

1. If the list `{`⟨*audiences*⟩`}` does *not* start with the keyword "-", it defines the audiences, which will see the corresponding content. All other audiences will not see it. For example, the command

   ```
   \showto{execs, devs}{TEXT}
   ```

   makes TEXT visible only to execs and devs, but not to anybody else among listed audiences.

2. If the list `{`⟨*audience*⟩`}` starts with the keyword "-", the logic is inverted: the audiences from the list will *not* see the contents. For example, the command

   ```
   \showto{-, execs, devs}{TEXT}
   ```

   makes the text not visible to execs and devs.

The scopes can be nested. For example, in the following situation TEXT 1 will be visible only to devs and execs, while TEXT 2 only to devs.

```
\begin{shownto}{devs,execs}
  TEXT 1
```

```
  \begin{shownto}{devs}
    TEXT 2
  \end{shownto}
\end{shownto}
```

There is, however, an important limitation: by nesting scopes you can hide a part of the document, but you cannot uncover it if it is hidden for the current audience by the enclosing scope. In other hands, the following makes TEXT 2 invisible to everybody:

```
\begin{shownto}{execs}
  TEXT 1
  \begin{shownto}{devs}
    TEXT 2
  \end{shownto}
\end{shownto}
```

2.3 Commands and environments

`\showto` The command `\showto{`⟨*audience*⟩`}{`⟨*contents*⟩`}` makes the `{`⟨*contents*⟩`}` visible (or invisible) to the current audience according to the first argument, as describes in Section 2.1.

`\Footnote` The macro `\Footnote{`⟨*audience*⟩`}{`⟨*footnote*⟩`}` creates a footnote with the visibility determined by the `{`⟨*audience*⟩`}` argument.

`shownto` The environment `\begin{shownto}{`⟨*audience*⟩`}`...`\end{shownto}` makes the contents visible or invisible according to its argument.

`Section` `Subsection` `Subsubsection` `Paragraph` `Subparagraph` Environments `Section`, `Subsection`, `Subsubsection`, `Paragraph`, `Subparagraph` define section-like units with limited visibility. They have the structure like `\begin{Section}{`⟨*audience*⟩`}[`⟨*short title*⟩`]{`⟨*long title*⟩`}`...`\end{Section}`, foe example

```
\begin{Section}{execs}[Executive comments]{Executive comments for
the document}
  Text
\end{Section}
```

2.4 Verbatim text

Due to the way verbatim is implemented in LaTeX, it is currently impossile to directly put `\verb` and `verbatim` inside visibility scopes. There are several workarounds shown in the example document:

1. You can put parts of the text with verbatim constructions in separate files and `\input` them inside visibility scopes.

2. You can use `\SaveVerbatim` and `\UseVerbatim` from *fancyvrb* package.

3. The command `\path` from *hyperref* package can be used for short verbatim-like fragments.

2.5 Extensions

`\DefMultiaudienceCommand`

The macro `\DefMultiaudienceCommand{`⟨*new command*⟩`}{`⟨*old command*⟩`}` can add visibility argument to `{`⟨*old command*⟩`}`. It is assumed that `{`⟨*old command*⟩`}` has only one argument. For example, `\Footnote` command in Section 2.3 was defined as

```
\DefMultiaudienceCommand{\Footnote}{\footnote}
```

`\NewMultiaudienceSectionEnv`

Similarly a section-like command `{`⟨*command*⟩`}` can be converted into an environment with explicit visibility scope with the command `\NewMultiaudienceSectionEnv {`⟨*environment*⟩`}{`⟨*command*⟩`}`. For example, `Section` environment is defined as

```
\NewMultiaudienceSectionEnv{Section}{\section}
```

2.6 Combined audiences

Sometimes one wants to create a document for several audiences. You can do this using comma-separated lists in `\CurrentAudience`, for example,

```
\DefCurrentAudience{execs,admins}
```

Important: there should be no spaces in the definition. The definition `\DefCurrentAudience{execs, admins}` will *not* work!

The resulting audience is the union of the audiences in the definition. Thus a fragment is included if either:

1. The list of audiences for the fragment does not start with `-`, and at least one of the current audiences is mentioned in the list.
2. The list of audiences for the fragment starts with `-`, and none of the current audiences is mentioned in the list.

For example, suppose the current audience is `execs,admins`. Then the following is true:

1. `\showto{devs}{...}`—excluded.
2. `\showto{devs,admins}{...}`—included.
3. `\showto{-, devs}{...}`—included.
4. `\showto{-, devs, admins}{...}`—excluded.

3 Implementation

```
1 ⟨*style⟩
```

3.1 Switches and defaults

`\CurrentAudience` The default for `\CurrentAudience`

```
2 \providecommand*\CurrentAudience{default}
```

`\DefCurrentAudience` A little sugar around `\def`:

```
3 \def\DefCurrentAudience#1{\def\CurrentAudience{#1}}
```

`\if@MULTAU@shown` The main switch

```
4 \newif\if@MULTAU@shown
5 \@MULTAU@showntrue
```

`\if@MULTAU@include` Whether the current command is "include" or "exclude" type (i.e. whether it does *not* start with "-")

```
6 \newif\if@MULTAU@include
7 \@MULTAU@includetrue
```

3.2 Key-value interface

We use `xkeyval`

```
8 \RequirePackage{xkeyval}
```

`\KV@MULTAU@-` Normally we evaluate visibility according to the following algorithm:

1. Set visibility to FALSE.
2. If found current audience in the list, set visibility to TRUE.

The presense of "-" keys inverts the algorithm:

1. Set visibility to TRUE.
2. If found current audience in the list, set visibility to FALSE.

```
9 \define@key{MULTAU}{-}[]{\@MULTAU@showntrue\@MULTAU@includefalse}
```

`\SetNewAudience` Definition of a new audience. We create a new key that switches visibility on or off.

```
10 \def\SetNewAudience#1{%
11   \define@key{MULTAU}{#1}[]{%
12     \def\@MULTAU@currkey{#1}%
13     \@for\@tempa:=\CurrentAudience \do{%
14       \ifx\@MULTAU@currkey\@tempa\relax
15         \if@MULTAU@include
16            \@MULTAU@showntrue
17         \else
18            \@MULTAU@shownfalse
19         \fi%
20     \fi%
21 }}}
```

3.3 Main command

\showto The macro \showto is the heart of the package. Note that we cannot put the inside into a group since we want it to be transparent for things like counters.

```
\long\def\showto#1#2{\@MULTAU@shownfalse\@MULTAU@includetrue
  \setkeys{MULTAU}{#1}%
  \if@MULTAU@shown#2\fi
  \@MULTAU@showntrue\@MULTAU@includetrue}
```

3.4 Main environment

We need the macros from the environ package

```
\RequirePackage{environ}
```

\shownto And the environment

```
\NewEnviron{shownto}[1]{%
  \@MULTAU@shownfalse\@MULTAU@includetrue
  \setkeys{MULTAU}{#1}%
  \if@MULTAU@shown\BODY\fi}{\@MULTAU@showntrue\@MULTAU@includetrue}
```

3.5 Extensions

\DefVisibilityCommand Create a new visibility command

```
\def\DefMultiaudienceCommand#1#2{\long\def#1##1##2{\showto{##1}{#2{##2}}}}
```

\Footnote Special footnote command:

```
\DefMultiaudienceCommand{\Footnote}{\footnote}
```

\NewMultiaudienceSectionEnv2 Create a new visibility section environment

```
\def\NewMultiaudienceSectionEnv#1#2{%
  \NewEnviron{#1}[1]{%
    \@MULTAU@shownfalse\@MULTAU@includetrue
    \setkeys{MULTAU}{##1}%
    \if@MULTAU@shown\expandafter#2\BODY\fi}%
  {\@MULTAU@showntrue\@MULTAU@includetrue}}
```

Section A bunch of section-like commands
Subsection
Subsubsection
Paragraph
Subparagraph

```
\NewMultiaudienceSectionEnv{Section}{\section}
\NewMultiaudienceSectionEnv{Subsection}{\subsection}
\NewMultiaudienceSectionEnv{Subsubsection}{\subsubsection}
\NewMultiaudienceSectionEnv{Paragraph}{\paragraph}
\NewMultiaudienceSectionEnv{Subparagraph}{\subparagraph}
```

⟨/style⟩

Change History

v1.01

`\DefVisibilityCommand`: Renamed 7

`\NewMultiaudienceSectionEnv2`: Renamed 7

`\if@MULTAU@shown`: Renamed 6

`\shownto`: Renamed 7

`\showto`: Renamed 7

v1.03

`\DefCurrentAudience`: Added macro 6

v1.04

`\SetNewAudience`: Added facility for a list of current audiences . . 6

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; numbers in roman refer to the code lines where the entry is used.

Symbols

\@MULTAU@currkey 12, 14
\@MULTAU@includefalse 9
\@MULTAU@includetrue 7, 22, 25, 28, 30, 35, 38
\@MULTAU@shownfalse 18, 22, 28, 35
\@MULTAU@showntrue . 5, 9, 16, 25, 30, 38
\@for 13
\@tempa 13, 14

B

\BODY 30, 37

C

\CurrentAudience 2, *2*, 3, 13

D

\def . 3, 10, 12, 22, 31, 33
\DefCurrentAudience *2*, 3
\define@key 9, 11
\DefMultiaudienceCommand *5*, 31, 32
\DefVisibilityCommand 31
\do 13

E

\else 17
environments:
  Paragraph *4*
  Section *4*
  shownto *4*
  Subparagraph *4*
  Subsection *4*
  Subsubsection . . . *4*
\expandafter 37

F

\fi . . . 19, 20, 24, 30, 37
\Footnote *4*, 32
\footnote 32

I

\if@MULTAU@include 6, 15
\if@MULTAU@shown 4, 24, 30, 37
\ifx 14

K

\KV@MULTAU@- 9

L

\long 22, 31

N

\NewEnviron 27, 34
\newif 4, 6
\NewMultiaudienceSectionEnv *5*, 33, 39, 40, 41, 42, 43
\NewMultiaudienceSectionEnv2 33

P

\Paragraph 39
Paragraph (environment) *4*
\paragraph 42
\providecommand 2

R

\relax 14
\RequirePackage . . 8, 26

S

\Section 39
Section (environment) *4*
\section 39
\setkeys 23, 29, 36
\SetNewAudience . . *3*, 10
\shownto 27
shownto (environment) *4*
\showto *4*, 22, 31
\Subparagraph 39
Subparagraph (environment) *4*
\subparagraph 43
\Subsection 39
Subsection (environment) *4*
\subsection 40
\Subsubsection 39
Subsubsection (environment) *4*
\subsubsection 41