The ‘accessibility’ package for tagged PDF documents

Andy Clifton

2023-01-18

Summary

Structured and tagged PDFs are required to meet modern corporate and governmental standards for document accessibility. PDFs that are created with core LaTeX are not tagged or structured, making it difficult to use LaTeX in a corporate or government environment. This document explains how the ***accessibility*** package can be used with LaTeX to prepare documents that pass such tests.

This document is also intended to be used as a test case as it contains most of the elements of a technical LaTeX document, including custom formatting, custom fonts, complex document structures, lists, equations, figures, tables, and code listings amongst others.

Table of Contents

List of Figures

List of Tables

1 Why is tagging required?

LaTeX is the de-facto standard for scientific publishing. LaTeX is often preferred over WYSIWYG word processors for technical documents because of the relatively simple file format that can be shared across users on many different platforms, and the ease of formatting a document for journal publication.

However, one issue with using LaTeX is document *accessibility*. Accessibility refers to the ability of a document to be reformatted for easier reading by visually impaired readers, or to work with a screen reader. This in turn requires that a document has a logical and correct machine-readable structure.

Accessibility is important for documents produced by government-funded organisations and public-facing bodies. Since the US Congress passed the 1998 Section 508 Amendment to the Rehabilitation Act of 1973, it has been a requirement that all US federally-funded documents are accessible to people with disabilities. This requirement has parallels in other countries.

An accessible PDF has several characteristics:

- All of the document content has been tagged
- It is possible to define a reading order based on those tags

- Images and links are given alternate text descriptions
- Tables are tagged, so that the table structure can be established
- Unicode descriptions of all characters are required

A document that has these characteristics is often referred to as being '508 compliant'. As 508-compliance is often judged using automated tests on the *.pdf* file, there is no option to work around this by using careful text descriptions of figures, for example.

In this document, I explain how LaTeXcan be generated using the ***accessibility*** style file.

My goal is that this will be a 'living' document and template that can be updated as we gain new insight into this process.

2 Making an accessible document using the accessibility package

LaTeX does not prepare a structured PDF document directly. Instead, we use the ***accessibility*** package to do this for us. This generates a tagged PDF that passes most automated document tests.

To use the ***accessibility*** package, simply add it to your preamble towards the end:

```
\usepackage[options]{accessibility}
```

2.1 Options

The basic options are:

untagged: no information about the structure

tagged: PDF with structure information

flatstructure: creates a flat structure

highstructure: creates a layered structure.

This document has been produced using the `[tagged, highstructure]` options.

2.2 For more information

More information about the ***accessibility*** package is included in the in German-language documentation.

3 Other steps to making an accessible document

The ***accessibility*** package does not solve all of the problems with making an accessible document.

3.1 Alternative text

Alternative text, or 'Alt text', is a textual description of an equation, link or figure that can be used to replace the visual information in that element. Alt text can be added after the PDF is compiled using a PDF editor, but should ideally be included by the author.

The ***accessibility*** package includes an `\alt{}` environment, which adds alt text to the PDF structure. It has been included in the source of the next equation.

(a) A chick. **(b) Another chick**

Figure 1. Test images

$$a^2 + b^2 = c^2 \tag{1}$$

Alt text can also take the form of a tooltip or pop-up. A tooltip can be generated from within the source document using the **pdftooltip** environment from the ***pdfcomment*** package.

For example, the left-hand panel in Figure **??** has been labeled with a tool tip. `\alt{}` has been used on the right-hand image.

A tooltip has also been used on the next equation. Passing the pointer over the following equation should reveal a pop-up:

$$a^2 + b^2 = c^2 \tag{2}$$

3.2 Embedding fonts

One requirement for passing automated tests for accessibility is that fonts must be embedded in the the final PDF. You can check the PDF for embedded fonts using a PDF viewer. For example, in Adobe Acrobat Reader, look at the 'fonts' tag of the document properties. If any fonts are not shown as being an *embedded subset*, you need to try again.

Encapsulated postscript figures are particularly prone to having undefined fonts. Check by compiling your document in draft mode, and seeing if the fonts are still present in the output PDF. To fix this problem, you could consider changing the *.eps* file to a *.png*. If you wish to do this 'on the fly', you could use this approach in your preamble:

```
\usepackage{epstopdf}
\epstopdfDeclareGraphicsRule{.eps}%
        {png}%
        {.png}%
```

```
        {convert eps:\SourceFile.\SourceExt png:\OutputFile}
\AppendGraphicsExtensions{.png}
```

4 Environment tests

4.1 Tables

Table **??** is included for testing purposes. It lists some of the packages that are included in this demonstration document. These packages often call other packages, so this is not an exhaustive list, but it is enough to test if the ***accessibility*** package works.

Table 1. An incomplete list of packages loaded in this document

Package	functionality
nag	checks that packages are up to date and looks for bad habits in LaTeX code.
geometry	sets page size and margins
mathptmx	changes fonts
helvet	changes fonts
courier	changes fonts
amsfonts, amssymb	supplies fonts that are useful for mathematics
booktabs	
graphicx	graphics handling, including *.eps* figures (see Section **??**)
pdfcomment	tool-tips. Also calls the package ***hyperref***

5 A template

The code used to produce this document is available from accessibility/tests/article/instructions-EN.tex.

6 Known problems with the accessibility package

There are bound to be a few.

If you find any, please use GitHub's issue tracking to report these. You can find the current list of issues at accessibility/issues.

The following is a summary of known problems with solutions and work-arounds where known:

- Line numbers in listings are incremented by 2 and not all lines are numbered. No work-around at this time.
- Formatted text in section headings causes errors. Work-around: don't include formatted text in headings.

Acknowledgements

This document benefitted from contributions to the http://tex.stackexchange.com/ website.

Babett Schalitz produced the original ***accessibility*** package in 2007 (**?**). That package was oriented towards KOMA-script documents. It was not accepted by CTAN and was subsequently not available to the LaTeXcommunity.

Babett Schalitz provided me with a copy of the original ***accessibility*** package in May 2019 and asked me to take up maintenance with a goal of submitting it to CTAN. This document is intended to support that effort. I am extremely grateful for all of Babett's work!

The ***accessibility*** package has since been available on Github, where a community of people has provided suggestions, bug fixes, and other support. I am very grateful for their help in maintaining and updating ***accessibility***.