AcroTEX.Net

The xbmks package
Cross-document bookmarks

D. P. Story

Prepared: January 18, 2020

www.acrotex.net
Version 2.0.3, 2020/01/16

Table of Contents

1. Introduction

It has been more than a couple decades ago (counting back from 2018), I wrote two mathematics tutorials: *eCalculus* and *Algebra Review in Ten Lessons*. The tutorials consisted of a number of lessons, each lesson was in a separate PDF. The bookmarks of each lesson contained the table of contents *for the whole tutorial*. A student, in theory, could then jump from one lesson to another by selecting a topic of interest from the bookmarks. In the intervening years I have not seen a LaTeX package for merging the table of contents of a set of PDFs and merge them in this each member of the set. The `xbmks` package attempts to reproduce this feature.

2. Required packages and options

The only requirement is the `hyperref` package; there is a negative requirement, the `bookmark` package is not supported. If it is detected within a `xbmks` document, normal non-cross-document bookmarks are produced.

The only options are driver options, these are `dvips` (Acrobat Distiller or ps2pdf can be used as the PDF creator), `pdftex` (and `luatex`, which is treated the same as `pdftex`), and `xetex`. Of course, this is a LaTeX package, so the 'la' versions of these applications need to be used. The package auto-detects `pdftex` and `xetex`, and `dvips` is the default, so there is actually no need to pass the driver option.

3. The preamble command `\xbmksetup`

The only user command is `\xbmksetup`:

```
\xbmksetup{%
    docbundle={⟨doc1⟩,⟨doc2⟩,...,⟨docn⟩},
    colors={int=⟨color⟩,ext=⟨color⟩},
    styles={intbf,extbf,intit,extit}
}
```

You have a collection of files (as specified by the `docbundle` key) that you want to merge the bookmarks together so they appear in all the files. However, the `\xbmksetup` only appears **in one and only one of the documents**, perhaps the one that you think of as the main file.[1] The argument of `\xbmksetup` are written to the file `xbmks.cfg` and input back into each of the files as they are compiled. This is needed so that across all files in the document bundle, they all obey the very same options.

Description of the key-values

`docbundle` (Optional) The value of this key, if present, is a comma-delimited list of base names that are a part of this document collection, or bundle. For example,

```
docbundle={lesson1,lesson2,lesson3,lesson4}
```

---

[1] The main file is the first one listed in the value of the `docbundle` key.

The order the bookmarks are listed in each file of the document bundle is the same order the files are listed in `docbundle`; in this above example, the bookmarks for `lesson1` are listed first, than those of `lesson2`, and so on.

The first document listed (`lesson1`) is where the `\xbmksetup` commend is placed. This file might be thought of as the 'main' file.

Important: It is extremely important to list the base names the using the same spelling and case as the `\jobname` command returns in each of the bundle collection. When each file is compiled, `xbmks` tries to determine which in the list of `docbundle` files is the one currently being compiled; it is necessary in order to create a internal jump or an external jump. If anything goes wrong, be sure to check the spelling of each file listed.

Empty or missing `docbundle` key. If this key is missing, then it is assigned a value of `\jobname`. As a result, normal bookmarks are generated for the document `\jobname`; however, as an extra benefit, the other keys are obeyed (`colors` and `styles`). See the sample file `stand-alone.tex` found in the `examples` file. This manual uses the `xbmks` package with,

```
\xbmksetup{colors={int=red},styles={intbf}}
```

`colors` (Optional) For any given document in the bundle, there are two types of links in the bookmarks, a link to an item in the current (or internal) document, or a link to an external document (within the bundle of documents).

- `int=`⟨*color*⟩, where ⟨*color*⟩ is an rgb color. This specifies the color of the link for an internal jump within the current document. Specify the color using the syntax `{[rgb]`⟨*r*⟩`,`⟨*g*⟩`,`⟨*b*⟩`}`; for example,

  ```
  colors={int={[rgb]{0,.6,0}}
  ```

  or, if the `xcolor` package is loaded, *named colors* may be used.
- `ext=`⟨*color*⟩, where ⟨*color*⟩ is an rgb color. This specifies the color of the link for an external jump to another member of the document bundle. Color specification is the same as for `int`:

  ```
  colors={int=red,ext=blue}
  ```

  In the above example, we assume `xcolor` is loaded and specify the colors accordingly.

  The default color. If you declare `int` (or `ext`) without a value, you get the default, which is the color key is not specified within the PDF document. The `color` key is optional, in which case, you get the default color. What is the default color? Well, its either white or black, depending on the Display Theme. The PDF viewer (Acrobat and Acrobat Reader DC automatically switch colors when the theme is changed).

  When you specify a color, be aware that what looks good in the *light theme* may not be so visible in a *dark theme*.

`styles` (Optional) The Adobe PDF viewer applications also support a bold and italics style. These can be specified for the internal and external documents.

- Internal styles (valueless) keys are `intbf` and `intit`; zero, one, or two of these may be specified,

```
styles={intbf}          % bold font
styles={intit}          % italics font
styles={intbf,intit}    % bold and italics font
```

- External styles (valueless) keys are `extbf` and `extit`; zero, one, or two of these may be specified,

```
styles={extbf}          % bold font
styles={extit}          % italics font
styles={extbf,extit}    % bold and italics font
```

Of course, all these keys are specified or not, in the `styles` key:

```
styles={intbf,extit}
```

Here, we specify bold font for the current document and an italics for an internal document.

Point of personal preference. After experimenting with various combinations of colors and styles with combinations of themes (light and dark) I prefer no color specified with bold font for the current document and plain font for any external document. I think it's important to make it clear in the bookmark panel which are internal and which are external. The bold font tells the same story independent of theme. Thus,

```
\xbmksetup{%
    docbundle={⟨doc1⟩,⟨doc2⟩,...,⟨docn⟩},
    styles={intbf}
}
```

seems to be a reasonable choice of key-values.

4. Creating bookmarks with other actions

The `hyperref` package provides commands (`\pdfbookmark`, `\currentpdfbookmark`, `\subpdfbookmark`, and `\belowpdfbookmark`) designed to create bookmarks that jump to a specified destination in the current document. The `xbmks` package now defines similar commands in which arbitrary actions may be defined.

```
\pdfbookmarkx[⟨level⟩]{⟨text⟩}[⟨KV-pairs⟩]{⟨name⟩}
\currentpdfbookmarkx{⟨text⟩}[⟨KV-pairs⟩]{⟨name⟩}
\subpdfbookmarkx{⟨text⟩}[⟨KV-pairs⟩]{⟨name⟩}
\belowpdfbookmarkx{⟨text⟩}[⟨KV-pairs⟩]{⟨name⟩}
```

If the optional ⟨*KV-pairs*⟩ argument is not present, the command behaves just like its `hyperref` counterpart; ⟨*name*⟩ is used to create a destination (or anchor) for the ordinary bookmark link. If ⟨*KV-pairs*⟩ is specified, no anchor is created, but ⟨*name*⟩ is used to associate the ⟨*action*⟩ with the bookmark.

Description of the commands

`\pdfbookmarkx` Creates a bookmark at level ⟨*level*⟩ in the outline tree hierarchy.

`\currentpdfbookmarkx` The command creates a bookmark at the current bookmark level in the outline tree.

`\subpdfbookmarkx` Reduces the current bookmark level by one, then creates the bookmark at that level. The reduced level is the new current bookmark level.

`\belowpdfbookmarkx` Creates a bookmark at one level below the current bookmark level without changing the value of the current bookmark level.

Description of the ⟨*KV-pairs*⟩ argument. The ⟨*KV-pairs*⟩ argument accepts up to three key-value pairs:

`action=`⟨*PDF-action*⟩`,color=`⟨*color*⟩`,style=`⟨`bf|it`⟩

Notice that `color` and `style` are in the singular, as opposed to the plural as they were in description of the key-value pairs for `\xbmksetup`, back on page 3.

`action=`⟨*PDF-action*⟩ ⟨*PDF-action*⟩ is raw PDF action code. I decided to just keep it simple. Consult Section 8.5 titled ‘Actions’, in particular, read Section 8.5.3 on ‘Action Types’ of the *PDF Reference Sixth Edition, Version 1.7*.[2] A general syntax for the ⟨*action*⟩ is,

`/S`⟨*action-type*⟩⟨*other-key-values*⟩

Common action-types are `/URI`, `/JavaScript`, `/Named`, and `/GoToR`. Below are some examples, the ones that appear in the demo files.

```
\belowpdfbookmarkx{http://www.acrotex.net}
  [action={/S/URI/URI(http://www.acrotex.net)}]{home}
\currentpdfbookmarkx{Current: Hello world!}
  [action={/S/JavaScript/JS(app.alert("Hello World!");)}]{bmk1}
\subpdfbookmarkx{Sub: First Page}[/S/Named/N/FirstPage]{bmk2}
\belowpdfbookmarkx{Go to doc1, page 1}
  [action={/S/GoToR/F(doc1.pdf)/D[1 /Fit]}]{gotor}
```

When eforms is loaded. The eforms package defines some helper commands for common action types, these are `\URI`, `\JS`, `\Named`, and `\GoToR`. The above examples are then written as,

```
\belowpdfbookmarkx{http://www.acrotex.net}
  [action={\URI{http://www.acrotex.net}}]{home}
\currentpdfbookmarkx{Current: Hello world!}
  [action={\JS{app.alert("Hello World!");}}]{bmk1}
\subpdfbookmarkx{Sub: First Page}
  [action={\Named{FirstPage}{]{bmk2}
\belowpdfbookmarkx{Go to doc1, page 1}
  [action={\GoToR/F(doc1.pdf)/D[1 /Fit]}]{gotor}
```

---

[2] https://www.adobe.com/devnet/pdf/pdf_reference_archive.html

`color=`⟨*color*⟩ The value of this key determines the color of the link. The value of ⟨*color*⟩ was described in the `colors` key. If this key is not specified, the value `int` or `ext` of the `colors` key is used, as appropriate. When the value of the `color` key is specified, the link receives the ⟨*color*⟩ across all documents in the bundle.

`style=`⟨`bf|it`⟩ The value of this key determines the style (`bf` – bold, `it` – italics); the keys may be used together `style={bf,it}` gives bold italics font. When `style` is not specified, the value of the `styles` key is used. When `style` is specified, the same style is assigned across all documents in the bundle.

```
\belowpdfbookmarkx{http://www.acrotex.net}
  [action={\URI{http://www.acrotex.net}},
   color=magenta,style=bf]{home}
```

This bookmark is colored magenta in bold across all documents in the bundle.

```
\belowpdfbookmarkx{http://www.acrotex.net}
  [action={\URI{http://www.acrotex.net}}]{home}
```

This bookmark takes on the colors and styles declared by the `colors` and `styles` keys.

5. Workflow

Given that you have a collection of files that are to contain a common set of bookmarks, how exactly do you do this? Basically, you treat them the same way as you would when you use the xy-hyper package, used for creating cross document links.

Steps to build the document bundle

1. Compile each document at least twice without deleting any auxiliary files, more if you are using xr-hyper; in particular, without deleting any OUT outline files created by hyperref.

   In terms of order of compilation, compile the main file first (the one that contains the `\xbmksetup` command); this will write the `xbmks.cfg` file and it will be available to the other files in the bundle as you compile them.

2. All files have been compiled twice, now compile them one more time so they can input the updated OUT files of the collection and the `xbmks.cfg` configuration file that contains your setup options for the whole collection.

3. If you are using pdflatex, lualatex, or xelatex, you are done; otherwise, convert each DVI file to PS using dvips. Finally, convert each PS file in the collection to PDF using Acrobat Distiller or ps2pdf.

4. Delete all auxiliary file, including `xbmks.cfg` if you wish.

If you are on Windows OS, you can use the AeB Builder utility to build the entire bundle of documents.[3] In that utility, you would select the External cross-references option, found on the user-interface.

Sample files. The demonstration files are found in the `examples` folder. The xcolor package is used, otherwise, the packages used are minimal.

Now, back to my retirement.

---

[3] http://www.acrotex.net/builders/