The phflplx package — include PDF graphics with hyperlinks

Philippe Faist *philippe.faist@bluewin.ch*

April 09, 2021

phflplx—A handy LaTeX package for including graphics defined via LaTeX source code files. Designed for use with the `ltxpdflinks` tool in order to include PDF graphics in documents with external or internal hyperlinks.

1 Introduction

This package is designed to be used in conjunction with the `ltxpdflinks` command-line utility to extract PDF links into `.lplx` files.

Usage:

1. Run `ltxpdflinks` on your PDF files before you compile your document:

   ```
   > ltxpdflinks myfigure.pdf
   ```

2. Use the following commands in your document preamble:

   ```
   \usepackage{phflplx}
   \DeclareGraphicsExtensions{.lplx,.pdf}
   ```

3. Drop a copy of `phflplx.sty` in the same folder as your document in case you don't have it installed user-wide or system-wide.

4. Enjoy!

In fact, LPLX files are simple LaTeX sources that are included directly by the graphics driver. The LPLX file typically includes the PDF itself using a lower-level

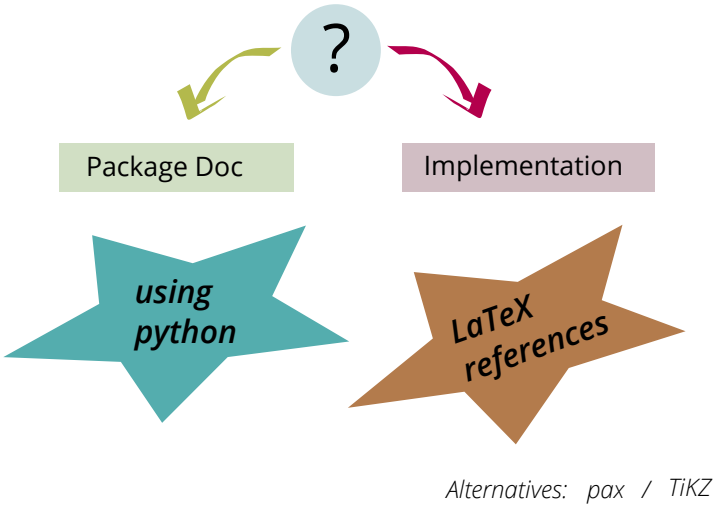

Figure 1: A diagram with clickable PDF hyperlinks, including internal links to other parts of the document.

graphics command and draws invisible hyperlinks on top of the graphics. (The LPLX file also takes care of resizing and cropping the graphics if necessary.)

Figure 1 provides a glimpse of what you can do with `ltxpdflinks` and phflplx.

■ 2 Implementation

Load some useful packages.

```
\RequirePackage{etoolbox}
```

Check that the user is loading the hyperref package! (We don't load it automatically to avoid issues of package loading order, link appearance, etc.)

```
\AtBeginDocument{%
  \@ifpackageloaded{hyperref}{}{%
    \PackageWarning{phflplx}{The package 'hyperref' was not loaded. You
      probably forgot to load it.}%
  }%
}
```

Declaring the graphics rule for the LPLX extension. Note that the LPLX file has the line `%%BoundingBox 0 0 <W> <H>` so that the size of the graphics can be parsed by the default graphics graphic size inspector, which is designed to parse `%%BoundingBox` commands in EPS files.

```
\DeclareGraphicsRule{.lplx}{lplx}{*}{}
```

Declare the driver functions for the graphics package internals.

```
\def\Ginclude@lplx#1{%
  \message{<#1>}%
  \input{#1}%
}
```

2.1 Commands called from the LPLX file

The contents of the LPLX file is wrapped around the `\LPLX` macro. The first argument is a dictionary `key1=value1,key2=value2,...` of meta-information about the current LPLX file format. Keys should include `version` (a basic file format version), `ltxpdflinksversion` (version of `ltxpdflinks` used to generate this file), `features` (a list of "features" the file provides, might add new "features" in the future). The "`bbox`" feature means that the LPLX file makes a call to `\lplxSetBbox` with the graphic bounding box information.

```
%%\long\def\LPLX#1#2{\begingroup #2\endgroup}
\let\LPLX\@gobble
```

In the contents of the LPLX data (second argument of `\LPLX`), we have a sequence of `\lplxAbcDef` commands in a declarative-style interface. The commands are the following.

`\lplxGraphic` Specify the original (PDF) graphic file name. Specify base file name (`#1`) and extension (`#2` including dot) separately

```
\def\lplxGraphic#1#2{%
  \def\lplxiGraphicBaseFname{#1}%
  \def\lplxiGraphicExt{#2}%
  \def\lplxiGraphicFname{#1#2}%
  \lplx@IncludeGraphics
}
```

`\lplxUserSpaceUnitLength` Set the base unit of the graphic. Width, height, coordinates, etc., are specified in this unit. Usually this is 1 bp = 1/72 in.

```
\def\lplxUserSpaceUnitLength#1{%
  \unitlength=#1\relax
}
```

`\lplxSetBbox` Declare what the bounding box of the graphic is. Currently the two first arguments should be `{0}{0}`.

```
\newdimen\lplxiBboxWcrdim
\newdimen\lplxiBboxHcrdim
\def\lplxSetBbox#1#2#3#4{%
  \def\lplxiBboxX{#1}%
```

```
  \def\lplxiBboxY{#2}%
  \def\lplxiBboxW{#3}%
  \def\lplxiBboxH{#4}%
  \ifdim\lplxiBboxX\p@>\z@\relax\lplx@bboxzerowarn\fi
  \ifdim\lplxiBboxX\p@<\z@\relax\lplx@bboxzerowarn\fi
  \ifdim\lplxiBboxY\p@>\z@\relax\lplx@bboxzerowarn\fi
  \ifdim\lplxiBboxY\p@<\z@\relax\lplx@bboxzerowarn\fi
}
\def\lplx@bboxzerowarn{%
  \PackageWarning{phflplx}{LPLX bounding box is not pinned at (0,0), not supported}%
}
```

`\lplxPicture` Gather all the links you want to place as an argument to a call to `\lplxPicture`.

```
\long\def\lplxPicture#1{%
  \lplx@SetupScaleAndBbox
  \lplx@a@DoScale{%
    \begin{picture}(\lplxvCropW,\lplxvCropH)(\lplxvCropX,\lplxvCropY)%
      #1%
    \end{picture}%
  }%
}
```

Finally, individual links are placed with `\lplxPutLink`. Usage is `\lplxPutLink{`⟨*x*⟩`}{`⟨*y*⟩`}{`⟨*w*⟩`}{`⟨*h*⟩`}{`⟨*hyperstartcmd*⟩`}{`⟨*hyperendtokens*⟩`}`. Here `{`⟨*hyperstartcmd*⟩`}` and `{`⟨*hyperendtokens*⟩`}` are any tokens that will be inserted immediately before and immediately after an invisible rule of the given width and height. The rule will be in curly braces, so it can be considered as a mandatory argument to the last macro token in `{`⟨*hyperstartcmd*⟩`}`. Don't forget to escape URL characters that have a LaTeX meaning that is very special (e.g. # → %23) because these characters are already in a macro argument and will have catcodes assigned before `\href` and friends get the chance to take care of catcodes etc. [Actually, some characters still work, like `^`, `_`, `&`, ... I'm not sure what determines this. (?)]

```
\def\lplxPutLink{%
  \ifGin@clip
    \expandafter\lplx@clipputlink
  \else
    \expandafter\lplx@doputlink
  \fi
}
```

A simple helper to percent-quote special characters in an URL.

```
{\catcode`\%=12\relax
  \gdef\lplx@percent{%}
}
\def\lplxHexChar#1{%
  \lplx@percent#1%
```

```
}
```

2.2 Internal Implementation Commands

`\lplx@IncludeGraphics` This command actually includes the underlying graphics. TODO: support for `\lplxiGraphicFname` in case the PDF base file name differs from the LPLX file base name? (Though that sounds like asking for trouble.)

```
\def\lplx@IncludeGraphics{%
  \edef\x{\noexpand\hbox to 0pt{%
      \noexpand\Ginclude@graphics{\Gin@base\lplxiGraphicExt}}}%
  \x
}
```

`\lplx@SetupScaleAndBbox` Utility to set up the appropriate command arguments to use for `\scalebox`, etc.

```
\def\lplx@noscale{1,!}
\def\lplx@exclam{!}
\def\lplx@a@DoScale{}%
\def\lplx@SetupScaleAndBbox{%
```

First, we locally define a macro `\lplx@a@DoScale{...}` that will generate the correct `\scalebox` call with the given contents, according to the requested size.

```
  \def\lplx@tmp{\Gin@scalex,\Gin@scaley}%
  \ifx\lplx@tmp\lplx@noscale%
    \def\lplx@a@DoScale{}%
  \else
    \ifx\Gin@scaley\lplx@exclam
      \edef\lplx@a@DoScale{\noexpand\scalebox{\Gin@scalex}}%
    \else
      \ifx\Gin@scalex\lplx@exclam
        \edef\lplx@a@DoScale{\noexpand\scalebox{\Gin@scaley}}%
      \else
        \edef\lplx@a@DoScale{\noexpand\scalebox{\Gin@scalex}[\Gin@scaley]}%
      \fi
    \fi
  \fi
```

Second, we need to take care of setting the bounding box correctly. Define the macros `\lplxvCropX` and `\lplxvCropY` which are the (X, Y) position of the lower left corner the part of the image we want to pick out, in user space units. Define `\lplxvCropW` and `\lplxvCropH` as the requested width & height of the subimage we want to use.

```
  \edef\lplxvCropX{\Gin@llx}%
  \edef\lplxvCropY{\Gin@lly}%
  \edef\lplxvCropW{\strip@pt\dimexpr\Gin@urx pt-\Gin@llx pt\relax}%
  \edef\lplxvCropH{\strip@pt\dimexpr\Gin@ury pt-\Gin@lly pt\relax}%
}
```

Tools to place links & clip them if necessary.

```
\def\lplx@doputlink#1#2#3#4#5#6{% x,y,w,h,hyperstart,hyperend
  \put(#1,#2){#5{\phantom{\rule{#3bp}{#4bp}}}#6}%
}%
\newdimen\lplx@tmpx
\newdimen\lplx@tmpy
\newdimen\lplx@tmpw
\newdimen\lplx@tmph
\def\lplx@clipputlink#1#2#3#4#5#6{% x,y,w,h,hyperstart,hyperend
```

Some notes: 1) We use "pt" as dummy unit of measure here just to do the floating point arithmetic and we use `\strip@pt` at the end. 2) Here `\lplx@maybeskip` serves as a flag that if set, asserts the link was entirely cropped out of the picture. Initially it expands to an empty string but when set it expands to "`\p@<\z@\relax`" (= "`1pt < 0pt`"), so it can be placed in front of all `\ifdim`'s so that they are skipped if the link was determined to be out of the picture.

```
  \def\lplx@maybeskip{}%
  \def\lplx@setskip{\def\lplx@maybeskip{\p@<\z@\relax}}%
  \lplx@tmpx=#1pt\relax
  \lplx@tmpy=#2pt\relax
  \lplx@tmpw=#3pt\relax
  \lplx@tmph=#4pt\relax
  \ifdim\lplx@maybeskip\lplx@tmpx<\lplxvCropX\p@\relax
    \ifdim\dimexpr\lplx@tmpx+\lplx@tmpw>\lplxvCropX\p@\relax
      \lplx@tmpw=\dimexpr\lplx@tmpx+\lplx@tmpw-\lplxvCropX\p@\relax
      \lplx@tmpx=\lplxvCropX\p@\relax
    \else
      \lplx@setskip
    \fi
  \fi
  \ifdim\lplx@maybeskip\dimexpr\lplx@tmpx+\lplx@tmpw
        >\dimexpr\lplxvCropX\p@+\lplxvCropW\p@\relax
    \ifdim\lplx@tmpx<\dimexpr\lplxvCropX\p@+\lplxvCropW\p@\relax
      \lplx@tmpw=\dimexpr\lplxvCropX\p@+\lplxvCropW\p@-\lplx@tmpx\relax
    \else
      \lplx@setskip
    \fi
  \fi
  \ifdim\lplx@maybeskip\lplx@tmpy<\lplxvCropY\p@\relax
    \ifdim\dimexpr\lplx@tmpy+\lplx@tmph>\lplxvCropY\p@\relax
      \lplx@tmph=\dimexpr\lplx@tmpy+\lplx@tmph-\lplxvCropY\p@\relax
      \lplx@tmpy=\lplxvCropY\p@\relax
    \else
      \lplx@setskip
    \fi
  \fi
  \ifdim\lplx@maybeskip\dimexpr\lplx@tmpy+\lplx@tmph
        >\dimexpr\lplxvCropY\p@+\lplxvCropH\p@\relax
    \ifdim\lplx@tmpy<\dimexpr\lplxvCropY\p@+\lplxvCropH\p@\relax
```

```
      \lplx@tmph=\dimexpr\lplxvCropY\p@+\lplxvCropH\p@-\lplx@tmpy\relax
    \else
      \lplx@setskip
    \fi
  \fi
  \ifdim\lplx@maybeskip\p@>\z@\relax
    \edef\x{\noexpand\lplx@doputlink%
      {\strip@pt\lplx@tmpx}{\strip@pt\lplx@tmpy}%
      {\strip@pt\lplx@tmpw}{\strip@pt\lplx@tmph}}%
    \x{#5}{#6}%
  \fi
}%
```

Change History

v0.1

General: Initial version 1

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.