The icon-appr Package

D. P. Story
Email: dpstory@acrotex.net

processed June 27, 2020

Contents

1 ⟨*package⟩

1 Required packages

```
2 \RequirePackage{xkeyval}
3 \RequirePackage{ifpdf}[2006/02/20]
4 \RequirePackage{ifxetex}[2006/08/21]
5 \RequirePackage{ifluatex}
6 \RequirePackage{eforms}[2018/11/10]
7 \RequirePackage{graphicx}
8 \@ifundefined{ifpdfmarkup}{\newif\ifpdfmarkup}{}\pdfmarkupfalse
9 \ifpdf\else\ifxetex\else\pdfmarkuptrue\fi\fi
10 \ifpdfmarkup\else
11 \def\ReqPkg{\RequirePackage{datatool}}\expandafter
12 \ReqPkg\fi
```

2 Introduction

The \specials and primitives of pdflatex, lualatex, and xelatex are used to provide support for inserting icon appearances into a button form field. We require eforms (2018/11/10) to create the form fields because the form support by hyperref does not provide the entries in a button field to create icon appearances. This package also supports icon appearances for dvips and dvipsone when the aeb_pro package is used and Acrobat Distiller or ps2pdf is used as the PDF creator, but in both cases, Adobe Acrobat must be the primary PDF viewer.[1]

3 Utility commands

3.1 Macros based on the **datatool** package

This version (2020/06/05 v1.2) of icon-appr attempts create the **AP** in the **Names** dictionary of the PDF catalog. A problem arose when I read from the PDF specification that the entries must be listed *alphabetically*. After a quick search of the internet for LaTeX packages that can alphabetize a list, came across a tex.stackexchange.com page, which solved the problem for me. In the lines below, the code from that page is modified to do the task I need it to do.

```
\newtoks\IA@IconsAlpha \IA@IconsAlpha={}
\let\IA@Icons\@empty
\let\IA@Icons@pdfmark\@empty
\let\IA@Icons@pdfmark@Names\@empty
\newcommand{\sortitem}[2][\relax]{%
  \DTLnewrow{list}% Create a new entry
  \ifx#1\relax
    \DTLnewdbentry{list}{sortlabel}{#2}%
  \else
    \DTLnewdbentry{list}{sortlabel}{#1}%
  \fi%
  \DTLnewdbentry{list}{description}{#2}%
}
\newenvironment{sortedlist}{%
  \DTLifdbexists{list}{\DTLcleardb{list}}{\DTLnewdb{list}}%
}{%
  \DTLsort{sortlabel}{list}% Sort list
  \DTLforeach*{list}{\theDesc=description}{\theDesc}%
}
\def\IA@sortitem#1{\sortitem[#1]{\IA@sortitemArgii{#1}}}
\def\IA@sortitemArgii#1{\edef\z{\global
  \IA@IconsAlpha={\the\IA@IconsAlpha (#1) \eq@relRef{\csOf{#1}}}}\z}
```

3.2 Other preliminaries

[1] This means the newly created PDF must be opened with Acrobat and saved before the file is ready for general use.

```
\def\x@namedef#1{\expandafter\xdef\csname #1\endcsname}
\def\heightOf#1{\@nameuse{#1heightOf}}
\def\widthOf#1{\@nameuse{#1widthOf}}
\newif\if@bEmbed \@bEmbedfalse
\newif\if@EmbedIconUsed \@EmbedIconUsedfalse
\newif\if@EmbedEPSUsed \@EmbedEPSUsedfalse
```

`\csOf{`⟨*name*⟩`}` Use `\csOf` to expand a control name.

```
\@ifundefined{csOf}{\let\csOf\@nameuse}{}
\let\IA@CreateImage@xsp\@empty
\let\IA@EndEmbedding@aebpro\relax
```

4 The main commands

```
\define@key{IAKVfamily}{name}[]{\def\IAKVfamily@name{#1}}
\define@key{IAKVfamily}{hyopts}[]{\def\IAKVfamily@hyopts{#1}}
\define@key{IAKVfamily}{cipresp}[]{\def\IAKVfamily@presp{#1}}
\define@key{IAKVfamily}{cipostsp}[]{\def\IAKVfamily@postsp{#1}}
\let\IAKVfamily@name\@empty
\let\IAKVfamily@hyopts\@empty
\let\IAKVfamily@presp\@empty
\let\IAKVfamily@postsp\@empty
```

4.1 The `embedding` environment

`embedding` The `embedding` environment encloses are embedding commands (`\embedIcon` and `\embedEPS`).

```
\newenvironment{embedding}{\def\IA@mark{[\space}\global\@bEmbedtrue
  \@ifpackageloaded{graphicxsp}{\let\IA@embedEPS@save\embedEPS
    \let\embedEPS\IA@embedEPS}{\let\embedEPS\IA@embedEPS@null}%
}{\global\@bEmbedfalse
```

(2020/06/05) In this, the `\end{embedding}` part of the environment, one or more `\embedIcon` commands have been inserted in the `\begin{embedding}` part of the environment. A list `\IA@Icons` is built, beginning with `\begin{sortedlist}`. Here, we close that environment. Then, if a dvips->distiller workflow *is not being used*, we close the `sortedlist` environment, then expand `\IA@Icons`, which should alphabetize the icon names, and leave results in the token register `\IA@IconsAlpha`, which we insert into the **AP** dictionary into the **Names** dictionary of the PDF catalog. The results looks like this (taken from one of my test files):

```
<</AP<</Names[(girl)27 0 R(mani)36 0 R(scot)41 0 R]>> >>
```

Note the entries of the **Names** dictionary of the **AP** dictionary are listed in alphabetical order along with their respective indirect reference.

```
  \g@addto@macro\IA@Icons{\end{sortedlist}}
```

We expand `\IA@Icons` only if not a pdfmark workflow, this is were the list of icon names is sorted.

```
  \ifpdfmarkup\else\expandafter\IA@Icons\fi
```

Write the **AP** dictionary to the **Names** dictionary for each driver.

```
\ifluatex
  \ifx\IA@Icons\@empty\else
    \pdfextension names{/AP <</Names[\the\IA@IconsAlpha]>>}\fi
\else\ifpdf
    \ifx\IA@Icons\@empty\else
      \immediate\pdfnames{/AP <</Names[\the\IA@IconsAlpha]>>}\fi
\else\ifxetex
    \ifx\IA@Icons\@empty\else
      \immediate\@pdfm@mark{obj @iconnames %
        << /Names [\the\IA@IconsAlpha] >> }%
      \@pdfm@mark{put @names %
        << /AP @iconnames >> }\fi
\fi\fi\fi
\gdef\IA@EndEmbedding{\IA@EndEmbedding@aebpro
\IA@CreateImage@xsp}\aftergroup\IA@EndEmbedding
}
```

The commands `\embedIcon` and `\embedEPS` are placed inside the `embedding` environment.

4.2 The `\embedIcon` command

`\embedIcon[`⟨*KVPairs*⟩`]{`⟨*path*⟩`}` This is the main embedding command of the package. An example of use is given below, as taken from `icons-appr-exmpl1.tex`.

```
\begin{embedding}
   \embedIcon[name=mani]{graphics/man1.pdf}
   \embedIcon[name=girl]{graphics/girl.pdf}
   \embedIcon[name=scot]{graphics/scot.pdf}
\end{embedding}
```

The `name` key is required for non-pdfmark drivers, its value must consist only of letters (or possibly non-active characters). The name value, say 'mani', is made into a control sequence, which may be referred to by `\mani` or `\csOf{mani}`.

The syntax for dvips/dvipsone is more complicated, we refer the user to the sample file `icons-appr-pb.tex`.

Initially, we test whether `\embedIcon` is enclosed in the `\bEmbeddeind`/`\eEmbedding` command pair. If yes, we fork off to any of five versions of `\embedIcon`, internally named `\IA@embedIcon`: (1) one for lualatex; (2) one for pdftex; (3) one for xelatex; (4) one for dvips with aeb_pro; and (5) one for the otherwise case, it does nothing but does avoids stopping the tex engine.

```
\newcommand{\embedIcon}{\global\@EmbedIconUsedtrue
  \if@bEmbed\def\IA@next{\IA@embedIcon}\else
    \PackageWarning{icon-appr}{\string\embedIcon\space commands must be
    enclosed\MessageBreak in the embedding environment}%
    \def\IA@next{\IA@embedIcon@null}\fi
  \IA@next}
\def\IA@handle@name@empty#1{{\count0=\aeb@image@cnt
```

```
    \advance\count0by1\relax\xdef
    \aeb@image@cnt{\the\count0 }}%
    \edef\IAKVfamily@name{aebImage\aeb@image@cnt}%
    \PackageWarning{icon-appr}{%
    The 'name' key is not specified,\MessageBreak
    will assign a name of '\IAKVfamily@name'\MessageBreak
    for the icon #1}}
```

Requirements of package not met. This version makes a necessary definition to avoid stopping the tex engine.

```
\newcommand{\IA@embedIcon@null}[2][]{%
  \begingroup
  \setkeys*{IAKVfamily}{#1}%
  \ifx\IAKVfamily@name\@empty\IA@handle@name@empty\fi
  \x@namedef{\IAKVfamily@name widthOf}{0pt}%
  \x@namedef{\IAKVfamily@name heightOf}{0pt}%
  \x@namedef{\IAKVfamily@name}{null}%
  \endgroup}
\let\IA@embedEPS@null\IA@embedIcon@null
\@ifundefined{aeb@image@cnt}{\def\aeb@image@cnt{0}}{}
\ifluatex
```

The lualatex driver. The two lualatex commands `\saveboxresource` and `\lastsavedboxresourceindex` are used.

```
\protected\def\pdfnames{\pdfextension names }
\newcommand{\IA@embedIcon}[2][]{\begingroup
  \setkeys*{IAKVfamily}{#1}%
  \ifx\IAKVfamily@name\@empty\IA@handle@name@empty{#2}\fi
  \toks@={\setbox\@tempboxa\hbox\bgroup\includegraphics}%
  \edef\x{\the\toks@[\IAKVfamily@hyopts]{#2}\egroup}\x
  \x@namedef{\IAKVfamily@name widthOf}{\the\wd\@tempboxa}%
  \x@namedef{\IAKVfamily@name heightOf}{\the\ht\@tempboxa}%
  \immediate\saveboxresource\@tempboxa
  \x@namedef{\IAKVfamily@name}
    {\the\lastsavedboxresourceindex}%
```

(2020/06/05) Entry in the icons array

```
  \ifx\IA@Icons\@empty
    \g@addto@macro\IA@Icons{\begin{sortedlist}}\fi
  \protected@edef\x{\noexpand\g@addto@macro\noexpand
    \IA@Icons{\protect\IA@sortitem{\IAKVfamily@name}}}\x
\endgroup}
\else\ifpdf
```

The pdflatex driver. The pdflatex commands `\pdfxform` and `\pdflastxform` are used.

```
\newcommand{\IA@embedIcon}[2][]{\begingroup
  \setkeys*{IAKVfamily}{#1}%
  \ifx\IAKVfamily@name\@empty\IA@handle@name@empty{#2}\fi
  \toks@={\setbox\@tempboxa\hbox\bgroup\includegraphics}%
  \edef\x{\the\toks@[\IAKVfamily@hyopts]{#2}\egroup}\x
```

```
  \x@namedef{\IAKVfamily@name widthOf}{\the\wd\@tempboxa}%
  \x@namedef{\IAKVfamily@name heightOf}{\the\ht\@tempboxa}%
  \immediate\pdfxform\@tempboxa
  \x@namedef{\IAKVfamily@name}{\the\pdflastxform}%
```

(2020/06/05) Entry in the icons array

```
  \ifx\IA@Icons\@empty
    \g@addto@macro\IA@Icons{\begin{sortedlist}}\fi
  \protected@edef\x{\noexpand\g@addto@macro\noexpand
    \IA@Icons{\protect\IA@sortitem{\IAKVfamily@name}}}\x
\endgroup}
\else\ifxetex
```

The xelatex driver. The xelatex `\specials` `bxobj`, `image`, and `exobj` are used. The `image` syntax does not support the concept of page for a PDF file, consequently, it always reads the first page.

```
\newcommand{\IA@embedIcon}[2][]{\begingroup
  \setkeys*{IAKVfamily}{#1}%
  \ifx\IAKVfamily@name\@empty\IA@handle@name@empty{#2}\fi
  \toks@={\setbox\@tempboxa\hbox\bgroup\includegraphics}%
  \edef\x{\the\toks@[\IAKVfamily@hyopts]{#2}\egroup}\x
  \x@namedef{\IAKVfamily@name widthOf}{\the\wd\@tempboxa}%
  \x@namedef{\IAKVfamily@name heightOf}{\the\ht\@tempboxa}%
  \x@namedef{\IAKVfamily@name}{@\IAKVfamily@name}%
  \@pdfm@mark{bxobj @\IAKVfamily@name\space
    width \the\wd\@tempboxa \space
    height \the\ht\@tempboxa \space}%
  \@pdfm@mark{image width \the\wd\@tempboxa \space
    height \the\ht\@tempboxa \space (#2)}%
  \@pdfm@mark{exobj}%
```

(2020/06/05) Entry in the icons array

```
  \ifx\IA@Icons\@empty
    \g@addto@macro\IA@Icons{\begin{sortedlist}}\fi
  \protected@edef\x{\noexpand\g@addto@macro\noexpand
    \IA@Icons{\protect\IA@sortitem{\IAKVfamily@name}}}\x
\endgroup}
\else
```

If not any of the previously discussed drivers, it must a **pdfmark**-type driver.

```
\@ifpackageloaded{aeb_pro}{%
```

For success in this case, we require the aeb_pro package; Acrobat needs to load and save the newly created PDF, created by either distiller or ps2pdf. The aeb_pro command `\declareImageAndPlacement` is used for this purpose. The command `\IA@EndEmbedding@aebpro` is redefined to input an `execJS` environment, which will get things going.

```
\newcommand{\IA@embedIcon}[2][1]{\begingroup
  \declareImageAndPlacement{path=#2,#1}\endgroup}
```

When aeb_pro is loaded and a pdfmark-type driver is used, \embedIcon uses \declareImageAndPlacement from aeb_pro (see documentation of that package). JavaScript methods embed the graphics files.

```
\def\IA@EndEmbedding@aebpro{\@bEmbedfalse\begingroup
  \IfFileExists{\jobname-exec.cut}{}
  {\immediate\openout\@auxout \jobname-exec.cut
  \immediate\write\@auxout{\string\begin{execJS}{IAexec}^^J%
  \string\insertPreDocAssembly^^J%
  \string\end{execJS}}}%
  \InputIfFileExists{\jobname-exec.cut}{}{}%
\endgroup}%
}{%
```

The final case, we \let \IA@embedIcon to \IA@embedIcon@null to let the document compile. A warning message is written to the log.

```
  \let\IA@embedIcon\IA@embedIcon@null
}
\fi\fi\fi
```

4.3 The \embedEPS command

\embedEPS[⟨*KVPairs*⟩]{⟨*path*⟩} When a **pdfmark**-type driver is being used, \embedEPS is \let to \IA@embedEPS. This command is used to embed EPS files.

```
\newcommand\IA@embedEPS[2][]{\ifx\IA@Icons@pdfmark@Names\@empty
  \global\let\IA@Icons@pdfmark@Names\@gobble\fi
  \global\@EmbedEPSUsedtrue
  \begingroup\let\x\noexpand
  \setkeys*{IAKVfamily}{#1}%\XKV@rm has remaining keys
  \ifx\IAKVfamily@name\@empty\IA@handle@name@empty\fi
  \edef\@tmpexp{\x
    \IA@embedEPS@save[\XKV@rm]{embed\IAKVfamily@name}{#2}}\@tmpexp
  \edef\@tmpexp{\x\begin{createImage}{\x
    \bboxOf{embed\IAKVfamily@name}}{\IAKVfamily@name}
    \ifx\IAKVfamily@presp\@empty\else\IAKVfamily@presp\space\fi
    \IA@mark{embed\IAKVfamily@name} /SP pdfmark
    \ifx\IAKVfamily@postsp\@empty\else\space\IAKVfamily@postsp\fi
    \x\end{createImage}}%
  \toks@=\expandafter{\@tmpexp}%
  \edef\@tmpexp{\x\g@addto@macro\x
    \IA@CreateImage@xsp{\the\toks@}}\@tmpexp
```

Create a list (\IA@Icons@pdfmark@Names) a list of all the names, this is used to create an array of icon names (see the end of the file). Also created is a token list of the icon names, this is employed in the definition of \IA@makeHiddenIconFlds at the end of this file.

```
  \edef\y{\x\g@addto@macro\x
    \IA@Icons@pdfmark@Names{,"\IAKVfamily@name"}}\y
  \edef\y{\x\g@addto@macro\x
    \IA@Icons@pdfmark{{\IAKVfamily@name}}}\y
```

```
  \endgroup
}
```

Some warning messages when things go wrong.

```
\def\IA@WarningMsgStr{%
    A pdfmark driver used - dvips,\MessageBreak
    something's not right however:
    \if@EmbedIconUsed
      \@ifpackageloaded{aeb_pro}{}{\MessageBreak
        Use of \string\embedIcon\space
        requires the\MessageBreak aeb_pro package,
        or possibly use\MessageBreak
        a non-pdfmark driver}\fi
  \if@EmbedEPSUsed
    \@ifpackageloaded{graphicxsp}{}{\MessageBreak
      \string\embedEPS\space
      requires the graphicxsp package}\fi
  }
\def\IA@WarningMsg{\PackageWarningNoLine{icon-appr}{\IA@WarningMsgStr}}
\ifpdfmarkup
  \AtBeginDocument{%
    \if@EmbedIconUsed
      \@ifpackageloaded{aeb_pro}{}{\ifx\IA@WarningMsg\relax\else
        \IA@WarningMsg\let\IA@WarningMsg\relax\fi}\fi
    \if@EmbedEPSUsed
      \@ifpackageloaded{graphicxsp}{}{\ifx\IA@WarningMsg\relax\else
      \IA@WarningMsg\let\IA@WarningMsg\relax\fi}\fi}%
\else
    \AtBeginDocument{\if@EmbedEPSUsed
      \PackageWarningNoLine{icon-appr}{The use of \string\embedEPS\space
      not allowed\MessageBreak
      with \ifluatex lualatex\else\ifpdf pdflatex\else xetex\fi\fi}
    \fi}%
\fi
```

`\IA@makeHiddenIconFlds` There is a final case of the workflow, when `\embedEPS` is used instead of `\embedIcon`. For **pdfmark**, the user has Acrobat, I'm assuming, and is not using ps2pdf without Acrobat. Here, we create a series of hidden `\pushButtons` fields each referencing one of the EPS embedded with `\embedEPS`; the creation of these fields is encapsulated into the `\IA@makeHiddenIconFlds`.

```
\def\IA@makeHiddenIconFlds{\if@EmbedEPSUsed
  \edef\x{\noexpand\@tfor\noexpand\iName:=\IA@Icons@pdfmark}\x\do{%
   \smash{\rlap{\pushButton[\BC{}\BG{}\F\FHidden\TP{1}\S{S}
    \I{\csOf{\iName}}]{IAhidden.\iName}{0bp}{0bp}}}%
  }%
  \fi
}
```

Insert `\IA@makdHiddenIconFlds` at the beginning of the document.

```
\AtBeginDocument{\IA@makeHiddenIconFlds}
```

Finally, a document JavaScript is inserted into the document that gets each of the fields created by `\IA@makeHiddenIconFlds`, and uses *Doc*`.addIcon()` and *Field*`.buttonGetIcon()` to "register" these icons. We then delete the fields after we are finished with them. `\IA@Icons@pdfmark@Names` will be either empty (`\embedEPS` is not used) or expands to a list of icon names, for each use of `\embedEPS`.

```
\ifpdfmarkup
\begin{insDLJS}{aicons}{icon-appr: Add icons to Catalog > Names > AP}
var aIconNames= new Array(\IA@Icons@pdfmark@Names);
for (i=0; i< aIconNames.length; i++) {
  var f=this.getField("IAhidden."+aIconNames[i]);
  if (f!=null) {
    try{ this.addIcon(aIconNames[i],f.buttonGetIcon());
    this.removeField("IAhidden."+aIconNames[i]); }catch(e){};
  }
}
\end{insDLJS}
\fi
⟨/package⟩
```

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

6 Change History

v1.1 (2020/06/04)

embedding: Entry in the icons array 4

v1.2.1 (2020/06/13)

General: \IA@Icons@pdfmark@Names is either empty or expands to a list of icon names. . . . 9

Replaced \ifpdfmarkup with \if@EmbedEPSUsed. Code should only be used if \embedEPS is used. 8