

The icon-appr Package

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

processed June 27, 2020

Contents

1	Required packages	1
2	Introduction	2
3	Utility commands	2
3.1	Macros based on the <code>datatool</code> package	2
3.2	Other preliminaries	2
4	The main commands	3
4.1	The embedding environment	3
4.2	The <code>\embedIcon</code> command	4
4.3	The <code>\embedEPS</code> command	7
5	Index	10
6	Change History	11
1	<code>*package</code>	

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 L^AT_EX 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.

```

13 \newtoks\IA@IconsAlpha \IA@IconsAlpha={}
14 \let\IA@Icons\@empty
15 \let\IA@Icons@pdfmark\@empty
16 \let\IA@Icons@pdfmark@Names\@empty
17 \newcommand{\sortitem}[2][\relax]{%
18   \DTLnewrow{list}% Create a new entry
19   \ifx#1\relax
20     \DTLnewbentry{list}{sortlabel}{#2}%
21   \else
22     \DTLnewbentry{list}{sortlabel}{#1}%
23   \fi%
24   \DTLnewbentry{list}{description}{#2}%
25 }
26 \newenvironment{sortedlist}{%
27   \DTLifdbexists{list}{\DTLcleardb{list}}{\DTLnewdb{list}}%
28 }{%
29   \DTLsort{sortlabel}{list}% Sort list
30   \DTLforeach*{list}{\theDesc=description}{\theDesc}%
31 }
32 \def\IA@sortitem#1{\sortitem[#1]{\IA@sortitemArgii{#1}}}
33 \def\IA@sortitemArgii#1{\edef\z{\global
34   \IA@IconsAlpha={\the\IA@IconsAlpha (#1) \eq@relRef{\csOf{#1}}}\z}

```

3.2 Other preliminaries

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

```

35 \def\x@namedef#1{\expandafter\xdef\csname #1\endcsname}
36 \def\heightOf#1{\@nameuse{#1heightOf}}
37 \def\widthOf#1{\@nameuse{#1widthOf}}
38 \newif\if@bEmbed \@bEmbedfalse
39 \newif\if@EmbedIconUsed \@EmbedIconUsedfalse
40 \newif\if@EmbedEPSUsed \@EmbedEPSUsedfalse

```

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

```

41 \@ifundefined{csOf}{\let\csOf\@nameuse}{}
42 \let\IA@CreateImage@xsp\@empty
43 \let\IA@EndEmbedding@aepro\relax

```

4 The main commands

```

44 \define@key{IAKVfamily}{name}[]{\def\IAKVfamily@name{#1}}
45 \define@key{IAKVfamily}{hyopts}[]{\def\IAKVfamily@hyopts{#1}}
46 \define@key{IAKVfamily}{cipresp}[]{\def\IAKVfamily@presp{#1}}
47 \define@key{IAKVfamily}{cipostsp}[]{\def\IAKVfamily@postsp{#1}}
48 \let\IAKVfamily@name\@empty
49 \let\IAKVfamily@hyopts\@empty
50 \let\IAKVfamily@presp\@empty
51 \let\IAKVfamily@postsp\@empty

```

4.1 The embedding environment

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

```

52 \newenvironment{embedding}{\def\IA@mark{[\space]\global\@bEmbedtrue
53 \ifpackageloaded{graphicxsp}{\let\IA@embedEPS@save\embedEPS
54 \let\embedEPS\IA@embedEPS}{\let\embedEPS\IA@embedEPS@null}%
55 }{\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.

```
56 \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.

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

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

```

58 \ifluatex
59   \ifx\IA@Icons\@empty\else
60     \pdfextension names{/AP <</Names [\the\IA@IconsAlpha]>>}\fi
61 \else\ifpdf
62   \ifx\IA@Icons\@empty\else
63     \immediate\pdfnames{/AP <</Names [\the\IA@IconsAlpha]>>}\fi
64 \else\ifxetex
65   \ifx\IA@Icons\@empty\else
66     \immediate\@pdfm@mark{obj @iconnames %
67       << /Names [\the\IA@IconsAlpha] >> }%
68     \@pdfm@mark{put @names %
69       << /AP @iconnames >> }\fi
70 \fi\fi\fi
71 \gdef\IA@EndEmbedding{\IA@EndEmbedding@aebpro
72 \IA@CreateImage@xsp}\aftergroup\IA@EndEmbedding
73 }
```

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.

```

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

```

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

```

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

```

88 \newcommand{\IA@embedIcon@null}[2] []{%
89 \begingroup
90 \setkeys*{IAKVfamily}{#1}%
91 \ifx\IAKVfamily@name\empty\IA@handle@name@empty\fi
92 \x@namedef{\IAKVfamily@name widthOf}{Opt}%
93 \x@namedef{\IAKVfamily@name heightOf}{Opt}%
94 \x@namedef{\IAKVfamily@name}{null}%
95 \endgroup}
96 \let\IA@embedEPS@null\IA@embedIcon@null
97 \ifundefined{aeb@image@cnt}{\def\aeb@image@cnt{0}}{}
98 \ifluatex

```

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

```

99 \protected\def\pdfnames{\pdfextension names }
100 \newcommand{\IA@embedIcon}[2] []{\begingroup
101 \setkeys*{IAKVfamily}{#1}%
102 \ifx\IAKVfamily@name\empty\IA@handle@name@empty{#2}\fi
103 \toks@={\setbox\@tempboxa\hbox\bgroup\includegraphics}%
104 \edef\x{\the\toks@[ \IAKVfamily@hyopts ]{#2}\egroup}\x
105 \x@namedef{\IAKVfamily@name widthOf}{\the\wd\@tempboxa}%
106 \x@namedef{\IAKVfamily@name heightOf}{\the\ht\@tempboxa}%
107 \immediate\saveboxresource\@tempboxa
108 \x@namedef{\IAKVfamily@name}
109 {\the\lastsavedboxresourceindex}%

```

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

```

110 \ifx\IA@Icons\empty
111 \g@addto@macro\IA@Icons{\begin{sortedlist}}\fi
112 \protected@edef\x{\noexpand\g@addto@macro\noexpand
113 \IA@Icons{\protect\IA@sortitem{\IAKVfamily@name}}}\x
114 \endgroup}
115 \else\ifpdf

```

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

```

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

```

```

121 \x@namedef{\IAKVfamily@name widthOf}{\the\wd\@tempboxa}%
122 \x@namedef{\IAKVfamily@name heightOf}{\the\ht\@tempboxa}%
123 \immediate\pdfxform\@tempboxa
124 \x@namedef{\IAKVfamily@name}{\the\pdflastxform}%
(2020/06/05) Entry in the icons array
125 \ifx\IA@Icons\@empty
126   \g@addto@macro\IA@Icons{\begin{sortedlist}}\fi
127 \protected@edef\x{\noexpand\g@addto@macro\noexpand
128   \IA@Icons{\protect\IA@sortitem{\IAKVfamily@name}}}\x
129 \endgroup}
130 \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.

```

131 \newcommand{\IA@embedIcon}[2][\begingroup
132   \setkeys*{\IAKVfamily}{#1}%
133   \ifx\IAKVfamily@name\@empty\IA@handle@name@empty{#2}\fi
134   \toks@={\setbox\@tempboxa\hbox\bgroup\includegraphics}%
135   \edef\x{\the\toks@[{\IAKVfamily\hyopts}{#2}\egroup]}\x
136   \x@namedef{\IAKVfamily@name widthOf}{\the\wd\@tempboxa}%
137   \x@namedef{\IAKVfamily@name heightOf}{\the\ht\@tempboxa}%
138   \x@namedef{\IAKVfamily@name}{@\IAKVfamily@name}%
139   \@pdfm@mark{bxobj @\IAKVfamily@name\space
140     width \the\wd\@tempboxa \space
141     height \the\ht\@tempboxa \space}%
142   \@pdfm@mark{image width \the\wd\@tempboxa \space
143     height \the\ht\@tempboxa \space (#2)}%
144   \@pdfm@mark{exobj}%
(2020/06/05) Entry in the icons array
145 \ifx\IA@Icons\@empty
146   \g@addto@macro\IA@Icons{\begin{sortedlist}}\fi
147 \protected@edef\x{\noexpand\g@addto@macro\noexpand
148   \IA@Icons{\protect\IA@sortitem{\IAKVfamily@name}}}\x
149 \endgroup}
150 \else

```

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

```

151 \@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.

```

152 \newcommand{\IA@embedIcon}[2][1]{\begingroup
153   \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.

```

154 \def\IA@EndEmbedding@aebpro{\@bEmbedfalse\beginngroup
155 \IfFileExists{\jobname-exec.cut}{}
156 {\immediate\openout\@auxout \jobname-exec.cut
157 \immediate\write\@auxout{\string\begin{execJS}{IAexec}^^J%
158 \string\insertPreDocAssembly^^J%
159 \string\end{execJS}}}%
160 \InputIfFileExists{\jobname-exec.cut}{-}{-}%
161 \endgroup}%
162 }%

```

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

```

163 \let\IA@embedIcon\IA@embedIcon@null
164 }
165 \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.

```

166 \newcommand\IA@embedEPS[2] []{\ifx\IA@Icons@pdfmark@Names\@empty
167 \global\let\IA@Icons@pdfmark@Names\@gobble\fi
168 \global\@EmbedEPSUsedtrue
169 \beginngroup\let\x\noexpand
170 \setkeys*{IAKVfamily}{#1}%\XKV@rm has remaining keys
171 \ifx\IAKVfamily@name\@empty\IA@handle@name@empty\fi
172 \edef\@tmpexp{\x
173 \IA@embedEPS@save[\XKV@rm]{embed\IAKVfamily@name}{#2}}\@tmpexp
174 \edef\@tmpexp{\x\begin{createImage}{\x
175 \bboxOf{embed\IAKVfamily@name}}{\IAKVfamily@name}
176 \ifx\IAKVfamily@presp\@empty\else\IAKVfamily@presp\space\fi
177 \IA@mark{embed\IAKVfamily@name} /SP pdfmark
178 \ifx\IAKVfamily@postsp\@empty\else\space\IAKVfamily@postsp\fi
179 \x\end{createImage}}%
180 \toks@=\expandafter{\@tmpexp}%
181 \edef\@tmpexp{\x\g@addto@macro\x
182 \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.

```

183 \edef\y{\x\g@addto@macro\x
184 \IA@Icons@pdfmark@Names{"\IAKVfamily@name"}}\y
185 \edef\y{\x\g@addto@macro\x
186 \IA@Icons@pdfmark{\IAKVfamily@name}}\y

```

```
187 \endgroup
188 }
```

Some warning messages when things go wrong.

```
189 \def\IA@WarningMsgStr{%
190   A pdfmark driver used - dvips,\MessageBreak
191   something's not right however:
192   \if@EmbedIconUsed
193     \ifpackageloaded{aeb_pro}{\MessageBreak
194       Use of \string\embedIcon\space
195       requires the\MessageBreak aeb_pro package,
196       or possibly use\MessageBreak
197       a non-pdfmark driver}\fi
198   \if@EmbedEPSUsed
199     \ifpackageloaded{graphicxsp}{\MessageBreak
200       \string\embedEPS\space
201       requires the graphicxsp package}\fi
202   }
203 \def\IA@WarningMsg{\PackageWarningNoLine{icon-appr}{\IA@WarningMsgStr}}
204 \ifpdfmarkup
205   \AtBeginDocument{%
206     \if@EmbedIconUsed
207       \ifpackageloaded{aeb_pro}{\ifx\IA@WarningMsg\relax\else
208         \IA@WarningMsg\let\IA@WarningMsg\relax\fi}\fi
209     \if@EmbedEPSUsed
210       \ifpackageloaded{graphicxsp}{\ifx\IA@WarningMsg\relax\else
211         \IA@WarningMsg\let\IA@WarningMsg\relax\fi}\fi}%
212 \else
213   \AtBeginDocument{\if@EmbedEPSUsed
214     \PackageWarningNoLine{icon-appr}{The use of \string\embedEPS\space
215     not allowed\MessageBreak
216     with \ifluatex lualatex\else\ifpdf pdflatex\else xetex\fi\fi}
217   \fi}%
218 \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`.

```
219 \def\IA@makeHiddenIconFlds{\if@EmbedEPSUsed
220   \edef\x{\noexpand\@tfor\noexpand\iName:=\IA@Icons@pdfmark}\x\do{%
221     \smash{\rlap{\pushButton[BC]{BG}{F\FHidden\TP{1}\S{S}
222       \I{\csOf{\iName}}]{IAhidden.\iName}{Obp}{Obp}}}%
223   }%
224   \fi
225 }
```

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

```
226 \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`.

```
227 \ifpdfmarkup
228 \begin{insDLJS}{aicons}{icon-appr: Add icons to Catalog > Names > AP}
229 var aIconNames= new Array(\IA@Icons@pdfmark@Names);
230 for (i=0; i< aIconNames.length; i++) {
231   var f=this.getField("IAhidden."+aIconNames[i]);
232   if (f!=null) {
233     try{ this.addIcon(aIconNames[i],f.buttonGetIcon());
234       this.removeField("IAhidden."+aIconNames[i]); }catch(e){};
235   }
236 }
237 \end{insDLJS}
238 \fi
239 \end{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.

Symbols	F
<code>\@EmbedEPSUsedfalse</code>	<code>\F</code> 221
<code>\@EmbedEPSUsedtrue</code>	<code>\FHidden</code> 221
<code>\@EmbedIconUsedfalse</code>	G
<code>\@EmbedIconUsedtrue</code>	<code>\g@addto@macro</code>
<code>\@auxout</code> 156, 157	.. 56, 111, 112, 126, 127, 146, 147, 181, 183, 185
<code>\@bEmbedfalse</code> 38, 55, 154	H
<code>\@bEmbedtrue</code> 52	<code>\heightOf</code> 36
<code>\@pdfmark</code> 66, 68, 139, 142, 144	I
<code>\@tempboxa</code> 103,	<code>\I</code> 222
105–107, 119, 121–123, 134, 136, 137, 140–143	<code>\IA@CreateImage@xsp</code> 42, 72, 182
<code>\@tmpexp</code> 172–174, 180–182	<code>\IA@embedEPS</code> 54, 166
A	<code>\IA@embedEPS@null</code> 54, 96
<code>\aeb@image@cnt</code> 80, 82, 83, 97	<code>\IA@embedEPS@save</code> 53, 173
<code>\aftergroup</code> 72	<code>\IA@embedIcon</code> 75, 100, 116, 131, 152, 163
<code>\AtBeginDocument</code> 205, 213, 226	<code>\IA@embedIcon@null</code> 78, 88, 96, 163
B	<code>\IA@EndEmbedding</code> 71, 72
<code>\bboxOf</code> 175	<code>\IA@EndEmbedding@aebpro</code> 43, 71, 154
<code>\BC</code> 221	<code>\IA@handle@name@empty</code> 80, 91, 102, 118, 133, 171
<code>\BG</code> 221	<code>\IA@Icons</code> 14, 56, 57, 59,
C	62, 65, 110, 111, 113, 125, 126, 128, 145, 146, 148
<code>\csOf</code> 3, 34, 41, 222	<code>\IA@Icons@pdfmark</code> 15, 186, 220
D	<code>\IA@Icons@pdfmark@Names</code> 16, 166, 167, 184, 229
<code>\declareImageAndPlacement</code> 153	<code>\IA@IconsAlpha</code> 13, 34, 60, 63, 67
<code>\DTLcleardb</code> 27	<code>\IA@makeHiddenIconFlds</code> 8, 219, 226
<code>\DTLforeach</code> 30	<code>\IA@mark</code> 52, 177
<code>\DTLifdbexists</code> 27	<code>\IA@next</code> 75, 78, 79
<code>\DTLnewdb</code> 27	<code>\IA@sortitem</code> 32, 113, 128, 148
<code>\DTLnewdbentry</code> 20, 22, 24	<code>\IA@sortitemArgii</code> 32, 33
<code>\DTLnewrow</code> 18	<code>\IA@WarningMsg</code> 203, 207, 208, 210, 211
<code>\DTLsort</code> 29	<code>\IA@WarningMsgStr</code> 189, 203
E	<code>\IAKVfamily@hyopts</code> 45, 49, 104, 120, 135
<code>\egroup</code> 104, 120, 135	<code>\IAKVfamily@name</code> 44, 48, 83, 86, 91–94, 102,
embedding (environment) <u>52</u>	105, 106, 108, 113, 118, 121, 122, 124, 128,
<code>\embedEPS</code> 53, 54, <u>166</u> , 200, 214	133, 136–139, 148, 171, 173, 175, 177, 184, 186
<code>\embedIcon</code> <u>74</u>	<code>\IAKVfamily@postsp</code> 47, 51, 178
environments:	<code>\IAKVfamily@presp</code> 46, 50, 176
embedding <u>52</u>	<code>\if@bEmbed</code> 38, 75
<code>\eq@relRef</code> 34	<code>\if@EmbedEPSUsed</code> 40, 198, 209, 213, 219
	<code>\if@EmbedIconUsed</code> 39, 192, 206
	<code>\IfFileExists</code> 155
	<code>\ifluatex</code> 58, 98, 216

<code>\ifpdf</code>	9, 61, 115, 216	<code>\rlap</code>	221
<code>\ifpdfmarkup</code>	8, 10, 57, 204, 227		
<code>\ifxetex</code>	9, 64, 130		
<code>\iName</code>	220, 222	S	
<code>\includegraphics</code>	103, 119, 134	<code>\S</code>	221
<code>\InputIfFileExists</code>	160	<code>\saveboxresource</code>	107
<code>\insertPreDocAssembly</code>	158	<code>\setbox</code>	103, 119, 134
		<code>\smash</code>	221
		<code>\sortitem</code>	17, 32
L			
<code>\lastsavedboxresourceindex</code>	109	T	
		<code>\theDesc</code>	30
O		<code>\TP</code>	221
<code>\openout</code>	156		
		W	
P		<code>\widthOf</code>	37
<code>\PackageWarning</code>	76, 84	<code>\write</code>	157
<code>\PackageWarningNoLine</code>	203, 214		
<code>\pdfextension</code>	60, 99	X	
<code>\pdflastxform</code>	124	<code>\x</code>	104, 112, 113, 120, 127, 128, 135, 147, 148, 169, 172, 174, 179, 181, 183, 185, 220
<code>\pdfmarkupfalse</code>	8	<code>\x@namedef</code>	35, 92–94, 105, 106, 108, 121, 122, 124, 136–138
<code>\pdfmarkuptrue</code>	9	<code>\XKV@rm</code>	170, 173
<code>\pdfnames</code>	63, 99		
<code>\pdfxform</code>	123	Y	
<code>\protected</code>	99	<code>\y</code>	183–186
<code>\protected@edef</code>	112, 127, 147		
<code>\pushButton</code>	221	Z	
		<code>\z</code>	33, 34
R			
<code>\ReqPkg</code>	11, 12		
<code>\RequirePackage</code>	2–7, 11		

6 Change History

v1.1 (2020/06/04)	empty or expands to a list of icon names. . . .	9
<code>embedding</code> : Entry in the icons array	Replaced <code>\ifpdfmarkup</code> with	
v1.2.1 (2020/06/13)	<code>\if@EmbedEPSUsed</code> . Code should only be	
General: <code>\IA@Icons@pdfmark@Names</code> is either	used if <code>\embedEPS</code> is used.	8