

The `aeb_pro` package  
as suggested by Jürgen Gilg

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

processed June 22, 2021

## Contents

<b>1</b>	<b>Alternate package name: <code>aeb-pro</code></b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>Declare Options</b>	<b>3</b>
3.1	AeB Control Central . . . . .	3
3.2	The <code>useacrobat</code> option . . . . .	5
3.3	<code>nopro</code> option . . . . .	5
3.4	<code>uselayers</code> option . . . . .	5
3.5	<code>gopro</code> option . . . . .	6
3.6	<code>attachsource</code> option . . . . .	6
3.7	<code>attachments</code> option . . . . .	6
3.8	The <code>linktoattachments</code> option . . . . .	6
<b>4</b>	<b>Package requirements</b>	<b>9</b>
<b>5</b>	<b>Support for PDF help</b>	<b>12</b>
<b>6</b>	<b>Support for OCG</b>	<b>13</b>
6.1	Initialization . . . . .	13
6.2	Supporting commands for OCG . . . . .	14
6.2.1	Document JavaScript in support of OCG . . . . .	14
6.2.2	Macros for creating rollovers . . . . .	15
6.2.3	Macros for layers . . . . .	15
6.2.4	The behavior of the rollovers . . . . .	17
6.3	In Support of Anime using Layers . . . . .	19
6.3.1	L <sup>A</sup> T <sub>E</sub> X Support . . . . .	19
6.3.2	JavaScript Support . . . . .	20

<b>7</b>	<b>Initial View</b>	<b>22</b>
7.1	Window Options	22
7.2	User Interface Options	22
7.3	Layout and Magnification	23
7.3.1	Navigation Tab (Page Mode)	23
7.3.2	Page Layout	23
7.3.3	Magnification	24
7.3.4	Open to page	24
<b>8</b>	<b>Fullscreen and Transitions</b>	<b>31</b>
8.1	<code>\setDefaultFS</code>	31
8.2	<code>\setPageTransition</code>	33
8.3	JS for Fullscreen Support	38
<b>9</b>	<b>Set Page Actions</b>	<b>39</b>
<b>10</b>	<b>Set Document Actions</b>	<b>44</b>
<b>11</b>	<b>In support of attachments</b>	<b>46</b>
11.1	Processing the <code>attachsource</code> and <code>attachments key</code>	46
11.2	Creating a PDF Package	50
<b>12</b>	<b>Support for linking to attachments.</b>	<b>52</b>
12.1	<code>\ahyperref</code> , <code>\ahyperlink</code> , <code>\ahyperextract</code>	52
<b>13</b>	<b><code>\labelName</code> and <code>\autolabelNum</code></b>	<b>58</b>
<b>14</b>	<b>Document Assembly Methods</b>	<b>61</b>
14.1	The <code>docassembly</code> environment	61
14.2	Supported Assembly JS API	61
14.3	Pre-docassembly Commands	66
14.3.1	Import and Place Images	66
14.3.2	Embed Multi-page Images	69
14.3.3	<code>\insertPreDocAssembly</code>	72
14.4	Placing an Image ( <code>\placeImage</code> )	72
14.5	Button and Ocg Animation (Anime)	73
14.5.1	Define <code>\btnAnime</code>	77
14.5.2	Define <code>ocgAnime</code>	78
<b>15</b>	<b>Index</b>	<b>86</b>
<b>16</b>	<b>Change History</b>	<b>97</b>

# 1 Alternate package name: aeb-pro

CTAN lists this package (aeb-pro) as aeb-pro, so we'll create a dummy package by that name.

```
1 <*altpkgname>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{aeb-pro}
4 [2021/02/04 v1.0 aeb-pro Alt-name (dps)]
5 \RequirePackage{xkeyval}
6 \DeclareOptionX*{\PassOptionsToPackage{\CurrentOption}{aeb_pro}}
7 \ProcessOptionsX\relax
8 %\RequirePackage{aeb_pro}

9 </altpkgname>
10 <*package>
```

## 2 Introduction

The aeb-pro package that access the ‘advanced’ features of the PDF specification. It uses various combinations of driver-dependent code, **pdfmark** code, and JavaScript methods and techniques that require Acrobat 7.0 Professional for document assembly. The package provides the **AeB Control Central**. Use the **AeB Control Central** to load all your AcroTeX packages in the correct order. See additional remarks in the titled paragraph **On the nature of the beast** below.

## 3 Declare Options

We declare some options for this package.

```
11 \@ifpackageloaded{docassembly}{%
12 \PackageError{aeb_pro}{The docassembly package is loaded.\MessageBreak
13     Functionality of this package is included in aeb\_pro.\MessageBreak
14     Please remove the docassembly package and recompile}{-}{-}
15 \RequirePackage{ifpdf}[2006/02/20]
16 \RequirePackage{ifxetex}[2006/08/21]
17 \RequirePackage{xkeyval}
```

Some standard flags used by this package.

```
18 \let\aebp@One=1 \let\aebp@Zero=0
19 \let\aebp@YES=y \let\aebp@N0=n
```

### 3.1 AeB Control Central

web	AeB control central is simply a set of options for including the other members of
exerquiz	the AeB family. This assures that the package are loaded in the correct order. The
dljslib	values of each of these option keys are the options of the packages they reference.
eforms	
insdljs	
eq2db	
aebxmp	
hyperref	
graphicxsp	
rmannot	
aebmlink	

(2016/12/09) Incorporate the version argument into package specification, for example,

```
\usepackage[web={designi,forpaper,req=2016/11/03}]{aeb_pro}
```

The version takes a key of `req`, as shown above.

`\ap@csarg` is a convenience macro for working with command names.

```
20 \def\ap@csarg#1#2{\expandafter#1\csname#2\endcsname}
```

`req` Define an `xkeyval` family `aebpreq` with a single key `req`. Later, we use this key to separate package options from the key-value `req=(release date)`.

```
21 \define@key{aebpreq}{req}[]{\def\eq@pkgReq{#1}}
```

A convenience command to save the value of package for later use.

```
22 \def\aebp@setCCOptions#1#2{#1=package #2=options}
```

```
23 \ap@csarg\edef{aeb@#1@opts}{#2}}
```

We not define options for `aeb_pro` of `web`, `exerquiz`, `eforms`, `insdljs`, `eq2db`, `aebxmp`, `hyperref`, `graphicxsp`, and `rmannot`.

```
24 \@tfor\aebp@pcks:={web}{exerquiz}{dljslib}%
```

```
25 {eforms}{insdljs}{eq2db}{aebxmp}{hyperref}{graphicxsp}{rmannot}{aeb_mlink}\do
```

```
26 {%
```

```
27 \ap@csarg\let{aeb@\aebp@pcks @req}\@empty
```

```
28 \ap@csarg\let{aeb@\aebp@pcks @opts}\relax
```

```
29 \edef\aebp@pack@def{\noexpand
```

```
30 \define@key{aeb_pro.sty}{\aebp@pcks}[]{%
```

```
31 \noexpand\aebp@setCCOptions{\aebp@pcks}{##1}}%
```

```
32 }\aebp@pack@def
```

```
33 }
```

`driver` We provide a convenient way to pass the `driver` used, `dvips` or `dvipsone`. If the user chooses the `web` option, then he/she might have a `web.cfg` file, so using this option may not be needed, in this case.

```
34 \let\aebp@passDriver\aebp@YES
```

```
35 \ifpdf\let\aebp@passDriver\aebp@NO\else
```

```
36 \ifxetex\let\aebp@passDriver\aebp@NO\fi\fi
```

```
37 \define@choicekey+{aeb_pro.sty}{driver}{%
```

```
38 {dvipsone,dvips,pdftex,dvipdfm,textures,%,
```

```
39 dvipdfmx,xetex}{%
```

```
40 \def\aebp@driver{#1}\ifx\aebp@driver\@empty\else
```

```
41 \PassOptionsToPackage{#1}{web}
```

```
42 \PassOptionsToPackage{#1}{exerquiz}
```

```
43 \def\aebp@PassToHYIns{\PassOptionsToPackage{#1}{hyperref}}
```

```
44 \PassOptionsToPackage{#1}{insdljs}}
```

```
45 \ifx\aebp@passDriver\aebp@YES
```

```
46 \expandafter\aebp@PassToHYIns\fi
```

```
47 \PassOptionsToPackage{#1}{eforms}
```

```
48 \PassOptionsToPackage{#1}{graphicxsp}
```

```
49 \fi
```

```
50 }\PackageError{aeb_pro}{Bad choice for the driver, permissible values
```

```
51 \MessageBreak are dvipsone, dvips, pdftex, dvipdfm,
```

```

52 textures,\MessageBreak dvipdfmx, and xetex}
53 {Try again with a permissible driver name}}
54 \let\aeB@driver\@empty

```

**On the nature of the beast** There are three types of commands (and environments) in this package: (1) commands that use driver dependent code, available to all workflows; (2) commands that depend on the **pdfmark** operator, available only to the dvips/Distiller workflow; and (3) commands that use ‘post-PDF creation’ methods that rely on the document author having the full Acrobat application.

For the workflow dvips/Distiller, all three code types are available; for whose authors that prefer pdflatex/lualatex/xelatex, but *do not have* the full Acrobat product, the commands of type (1) are available through the **nopro** option. Finally, for authors that prefer pdflatex/lualatex/xelatex and *do have* Acrobat, the commands of type (1) and (3) are available through the **useacrobat** option.

### 3.2 The useacrobat option

**useacrobat** If the document author has Acrobat, but wants to use pdflatex, xelatex, etc, we shall let them but under restricted conditions using the **useacrobat** option. Essentially, all of **aeB\_pro** is available except for any commands that use layers; hence, the **uselayers** and **ocganime** options are not supported.

```

55 \newif\if@useacrobat \@useacrobatfalse
56 \DeclareOptionX{useacrobat}{\@useacrobattrue}

```

### 3.3 nopro option

**nopro** The normally **aeB\_pro** will brings in all the code that would require the use of the Distiller and Acrobat Pro, this is the default. The **nopro** option gives access, for non-Distiller workflows, to the **AeB Control Center** and to all commands that do no involve layers or post-PDF creation methods.

```

57 \newif\if@gopro \@goprotrue
58 \DeclareOptionX{nopro}{\@useacrobattrue\@goprofalse}
59 \PassOptionsToPackage{nopro}{insdljs}

```

**Input aeBpro.def** We input **aeBpro.def** at the end of the package.

```

60 \def\aeB@InputAeBPro{\InputIfFileExists{aeBpro.def}%
61 {\PackageInfo{aeB_pro}{Inputting code for the pro option}}%
62 {\PackageInfo{aeB_pro}{Cannot find aeBpro.def.
63 Reinstall or refresh your file name database, as needed.}}}

```

### 3.4 uselayers option

**uselayers** The **uselayers** option activates the use of Optional Content Groups, a.k.a., layers.

```

64 \let\aeBp@uselayers=0
65 \DeclareOptionX{uselayers}{%
66 \ifx\aeBp@uselayers0\AtEndOfPackage{\aeB@InputAeBLayers}%

```

```

67 \global\let\aebp@uselayers=1\fi
68 }
69 \def\aeB@InputAeBLayers{\InputIfFileExists{aeblayers.def}%
70 {\PackageInfo{aeb_pro}{Inputting code for uselayers option}}%
71 {\PackageInfo{aeb_pro}{Cannot find aeblayers.def.
72 Reinstall or refresh your file name database.}}}

```

### 3.5 gopro option

**gopro** Some of the components of AeB have the pro option. Use `gopro` to pass the pro option to all components that take it.

```

73 \DeclareOptionX{gopro}{\PassOptionsToPackage{pro}{web}}

```

### 3.6 attachsource option

**attachsource** A simple interface, `attachsource` to attaching the source file(s) of `\jobname`.

```

74 \DeclareOptionX{attachsource}{\def\aeB@attachsource{#1}}
75 \let\aeB@attachsource\@empty

```

### 3.7 attachments option

**attachments** The `attachments` option is a general routine for attaching files other than the `\jobname`.

```

76 \DeclareOptionX{attachments}{\def\aeB@attachments{#1}}
77 \let\aeB@attachments\@empty

```

**optattachments** The two options, `optattachments` and `!optattachments`, set the Boolean switch `\ifoptattachments` to true or false, respectively. the default is false. When `!optattachments` is used, files are input into the source file, but also attached to the PDF when `optattachments` is used, and not attached when `!optattachments` is used.

```

78 \newif\ifoptattachments\optattachmentsfalse
79 \newif\ifoptattachmentsTaken\optattachmentsTakenfalse
80 \DeclareOptionX{optattachments}{%
81 \optattachmentsTaketrue
82 \optattachmentstrue}
83 \DeclareOptionX{!optattachments}{%
84 \optattachmentsTaketrue
85 \optattachmentsfalse}

```

### 3.8 The linktoattachments option

**linktoattachments** The option `linktoattachments` is a general routine for linking to attached PDF files.

```

86 \def\aeB@InputUnicodes{\InputIfFileExists{linkto.def}%
87 {\PackageInfo{aeb_pro}{Inputting code for the linktoattachments
88 option}}{\PackageWarning{aeb_pro}{Cannot find linkto.def.
89 Reinstall or refresh your file name database, as needed.}}}

```

```

90 \let\ae@InputUnicode@empty
(2015/08/06) Added code so that the childof is read after linktoattachments
91 \DeclareOptionX{linktoattachments}{%
92   \let\ae@InputUnicode@ae@InputUnicode
93 }
94 \def\ap@RequireUnicode{\RequirePackage{forms16be}}
95 \let\ae@childofInput@empty
96 \let\ae@childof@empty
97 \DeclareOptionX{childof}{\def\ae@childof{#1}\def\ae@childofInput{%
98   \InputIfFileExists{\ae@childof_xref.cut}{}}}%
99 \def\inputAttachmentRelatedFiles{%
100   \ae@InputUnicode
101   \ae@childofInput
102 }
103 \AtEndOfPackage{\inputAttachmentRelatedFiles}

```

**ocganime** When the ocganime option is taken, the code for ocg animation is included.

```

104 \let\inputCommonAnimeCode\relax
105 \DeclareOptionX{ocganime}{%
106   \ifx\ae@uselayers0\ExecuteOptionsX{uselayers}
107   \global\let\ae@uselayers=1\fi
108   \ifx\inputCommonAnimeCode\relax
109     \let\inputCommonAnimeCode\ae@inputCommonAnimeCode
110     \AtEndOfPackage{\inputCommonAnimeCode}\fi
111   \AtEndOfPackage{\inputOcgAnimeCode}
112 }

```

**btnanime** When the btnanime option is taken, the code for button animation is included.

```

113 \DeclareOptionX{btnanime}{%
114   \ifx\inputCommonAnimeCode\relax
115     \let\inputCommonAnimeCode\ae@inputCommonAnimeCode
116     \AtEndOfPackage{\inputCommonAnimeCode}\fi
117   \AtEndOfPackage{\inputBtnAnimeCode}
118 }
119 \def\inputBtnAnimeCode{\InputIfFileExists{btnanime.def}
120   {\PackageInfo{ae_pro}{inputting btnanime.def}}
121   {\PackageWarning{ae_pro}{cannot find btnanime.def}}}%
122 \def\inputOcgAnimeCode{\InputIfFileExists{ocganime.def}
123   {\PackageInfo{ae_pro}{inputting ocganime.def}}
124   {\PackageWarning{ae_pro}{cannot find ocganime.def}}}%
125 \def\ae@inputCommonAnimeCode{\InputIfFileExists{anime_support.def}
126   {\PackageInfo{ae_pro}{inputting anime_support.def}}
127   {\PackageWarning{ae_pro}{cannot find anime_support.def}}}%

```

We allow for a configuration file, usually of the form

```
\ExecuteOptionsX{driver=<driver>} % dvips, dvipsone, pdftex, xetex
```

For the last two, AeB Pro can only act as a central control of the AeB Family. If the driver being used (automatically detected) is `pdftex` or `xetex`, we use the `nopro` option and don't input `aeopro.cfg`.

```

128 \let\pathToBtnCtrlIcons\@empty
129 \let\ExecuteOptionsXSAVE\ExecuteOptionsX
130 \let\ExecuteOptionsX\@gobble
131 \InputIfFileExists{aebpro.cfg}{-}{-}
132 \let\ExecuteOptionsX\ExecuteOptionsXSAVE
133 \ifpdf\ExecuteOptionsX{driver=pdfTeX}\else
134   \ifxetex\ExecuteOptionsX{driver=xetex}\else
135   \ifundefined{!@tex@@@driver}{\ExecuteOptionsX{driver=dvips}}
136     {\ExecuteOptionsX{driver=dvipsone}}\fi\fi

```

### Process the options

```

137 \ProcessOptionsX
138 \edef\ap@restoreCats{%
139   \catcode'\noexpand\"=\the\catcode'\\"relax
140   \catcode'\noexpand\,=\the\catcode'\,\relax
141   \catcode'\noexpand\(\=\the\catcode'\(\relax
142   \catcode'\noexpand\!=\the\catcode'\!\relax
143 }
144 \@makeother\" \@makeother\, \@makeother\(\ \@makeother\!

```

**Set up variables** We set a switch to indicate we are using the **pdfmark** operator or not. Initially we assume so.

```

145 \ifundefined{ifpdfmarkup}{\newif{ifpdfmarkup}}
146 \pdfmarkuptrue

```

Added support for luatex .85. Moved this code to insdljs (2016/07/31); aeb\_pro requires insdljs so put it there.

```

\ifpdf\ifx\pdffeedback\@undefined\else
\def\pdfpageref{\pdffeedback pageref}\fi\fi

```

Set the variables for the case of dvips and dvipsone.

```

147 \def\aeB@testdriver{dvips}%
148 \ifx\aeB@testdriver\aeB@driver
149   \useacrobattrue\else
150 \def\aeB@testdriver{dvipsone}%
151 \ifx\aeB@testdriver\aeB@driver
152   \useacrobattrue\else

```

If not dvips or dvipsone, the user is not to use the **pdfmark** operator.

```

153   \pdfmarkupfalse
154 \fi\fi

```

### Summary of switch settings

- Distiller workflow: `\pdfmarkuptrue, \@useacrobattrue, \@goprotrue`
- non-Distiller/non-Acrobat workflow (the `nopro` option): `\pdfmarkupfalse, \@useacrobattrue, \@goprofalse`



- non-Distiller/Acrobat workflow (the useacrobat option): `\pdfmarkupfalse`, `\@useacrobattrue`, `\@goprotrue`

Some options are not valid when using the nopro, such as attachments.

```

155 \if@gopro\else
156 \let\aebp@isattach\aebp@NO
157 \ifx\aebp@attachsource\@empty\else\let\aebp@isattach\aebp@YES\fi
158 \ifx\aebp@attachments\@empty\else\let\aebp@isattach\aebp@YES\fi
159 \let\aebp@attachsource\@empty
160 \let\aebp@attachments\@empty
161 \optattachmentsfalse\optattachmentsTakenfalse
162 \let\inputAttachmentRelatedFiles\relax
163 \ifx\aebp@isattach\aebp@YES
164   \PackageWarningNoLine{aeb_pro}{The various attachment options
165     are not\MessageBreak supported with the nopro option,
166     ignoring them}\fi
167 \fi

```

**Disallow some options that use the pdfmark** We do what we say.

```

168 \ifpdfmarkup\else
169 \if\aebp@uselayers1
170   \def\aebp@InputAeBLayers{\PackageError{aeb_pro}
171     {The uselayers and ocanime options are not supported,\MessageBreak
172     with the useacrobat and nopro options,\MessageBreak
173     ignoring these options}{Remove dependence on layers,
174     or obtain\MessageBreak the full Acrobat product}}
175   \let\texHelpIndicator\@gobble
176   \let\addJStexHelpEnter\@gobble
177   \let\addJStexHelpExit\@gobble
178   \let\texHelp\@gobbletwo
179   \def\xBld#1\@eBld{}
180   \let\defineRC\@gobbletwo
181   \let\insertRC\@gobble
182   \let\definePR\@gobbletwo
183   \let\insertPR\@gobble
184   \let\proofRollovers\relax
185   \let\texPHProof\@gobbletwo
186   \newcommand\animeBld[1] [] {}
187   \def\DeclareAnime#1#2#3{}
188   \newcommand\backAnimeBtn[3] [] {}
189   \newcommand\clearAnimeBtn[3] [] {}
190   \newcommand\forwardAnimeBtn[3] [] {}
191 \fi\fi

```

## 4 Package requirements

AeB builds on the web package to a certain extent, hence requires insdljs.

```

192 \def\aebp@PkgWarningi#1{\ifpdfmarkup\else\PackageWarningNoLine{aeb_pro}

```

```

193   {The #1\space package requires Adobe Distiller,\MessageBreak
194     do not load this package}\fi}
195 \def\aeBPkgWarningii#1{\PackageWarningNoLine{aeb_pro}
196   {The #1\space package requires Adobe Distiller,\MessageBreak
197     ignoring #1\space as an option}}
198 \def\aeBPkgWarningiii#1{\PackageWarningNoLine{aeb_pro}
199   {The #1\space package requires Adobe Distiller or\MessageBreak
200     the useacrobat option (Acrobat required),\MessageBreak
201     ignoring #1\space as an option}}
202 \providecommand\getargsii[2]{\def\aeBargi{#1}\def\aeBargii{#2}}

\ap@RP is a command that computes the appropriate \PassOptionsToPackage
and \RequiredPackage commands

#1 aeb_pro required options, #2 = package name

203 \newcommand\ap@RP[2][\def\@rgi{#1}\def\@rgii{#2}%
204   \ifx\@rgi\@empty\else\edef\@rgi{\@rgi}\fi
205   \@ifnextchar[%]
206   {\ap@RP}{\ap@RP[]}}

\ap@@RP continues \ap@RP and is the one that formulates \PassOptionsToPackage
and \RequiredPackage commands

#1 = aeb_pro required data

207 \def\ap@@RP[#1]{\def\@rgiii{#1}%
208   \let\XKV@rm\@empty\let\eq@pkgReq\@empty
209   \@ifundefined{aeb@\@rgii @opts}
210   {\ifx\@rgi\@empty\else
211     \ap@csarg\edef{aeb@\@rgii @req}{#1}\fi
212   }{}}

Calculate \PassOptionsToPackage and its arguments

213   \edef\x{\noexpand
214     \setkeys*{aebpreq}{\@nameuse{aeb@\@rgii @opts}}}\x
215   \edef\eq@theseOpts{\XKV@rm}%
216   \ap@csarg\edef{aeb@\@rgii @req}{\eq@pkgReq}%
217   \edef\x{\noexpand
218     \PassOptionsToPackage{\eq@theseOpts}{\@rgii}}\x
219   }%

Construct \RequirePackage and its arguments

220   \ifx\@rgiii\@empty
221     \if!\@nameuse{aeb@\@rgii @req}!\else
222     \ap@csarg\edef{aeb@\@rgii @req}{[\@nameuse{aeb@\@rgii @req}]\fi
223   \else
224     \@ifl@t@r{#1}{\@nameuse{aeb@\@rgii @req}}%
225     {\if!\@nameuse{aeb@\@rgii @req}!\else
226     \ap@csarg\edef{aeb@\@rgii @req}%
227       {[\@nameuse{aeb@\@rgii @req}]\fi
228     }{\ap@csarg\edef{aeb@\@rgii @req}{[#1]}}%
229   \fi

```

```

230 \edef\x{\noexpand
231 \RequirePackage{\@rgii}\@nameuse{aeb@\@rgii @req}}\x
232 }
233 \@ifpackageloaded{web}{
234 {%
235 \ifundefined{aeb@web@opts}{\ap@RP{web}}%
236 }
237 \@ifpackageloaded{exerquiz}{
238 {%
239 \ifundefined{aeb@exerquiz@opts}{\ap@RP{exerquiz}}%
240 }
241 \@ifpackageloaded{hyperref}{
242 {%
243 \ifundefined{aeb@hyperref@opts}{\ap@RP{hyperref}}%
244 }
245 \@ifpackageloaded{eq2db}{
246 {%
247 \ifundefined{aeb@eq2db@opts}{\ap@RP{eq2db}}%
248 }
249 \@ifpackageloaded{dljslib}{
250 {%
251 \ifundefined{aeb@dljslib@opts}{\ap@RP{dljslib}}%
252 }
253 \@ifpackageloaded{eforms}{\let\execjs\aebp@YES}
254 {%
255 \ifundefined{aeb@eforms@opts}{
256 \ap@RP[execJS]{eforms}[2020/12/14]}%
257 }

```

load eforms before insdljs so eforms can properly pass options to insdljs

```

258 \@ifpackageloaded{insdljs}{\let\execjs\aebp@YES}
259 {%
260 \ap@RP[execJS]{insdljs}[2021/02/07]}%
261 }
262 \@ifpackageloaded{aebxmp}{\aebp@PkgWarningi{aebxmp}}
263 {%
264 \ifundefined{aeb@aebxmp@opts}{
265 {\edef\temp@exp{\ifpdfmarkup
266 \noexpand\ap@RP{aebxmp}\else
267 \if@useacrobat
268 \noexpand\ap@RP{aebxmp}[2016/05/12]\else
269 \noexpand\aebp@PkgWarningiii{aebxmp}\fi\fi}\temp@exp}
270 }
271 \@ifpackageloaded{graphicxsp}{\aebp@PkgWarningi{graphicxsp}}
272 {%
273 \ifundefined{aeb@graphicxsp@opts}{
274 {\edef\temp@exp{\ifpdfmarkup
275 \noexpand\ap@RP{graphicxsp}\else
276 \noexpand\aebp@PkgWarningii{graphicxsp}\fi}\temp@exp}
277 }

```

```

278 \@ifpackageloaded{rmannot}{\aebp@PkgWarningi{rmannot}}
279 {%
280   \ifundefined{aeb@rmannot@opts}{}
281   {\edef\temp@exp{\ifpdfmarkup\noexpand\ap@RP{rmannot}\else
282     \noexpand\aebp@PkgWarningii{rmannot}\fi}\temp@exp}
283 }
284 \@ifpackageloaded{aeb_mlink}{\aebp@PkgWarningi{aeb_mlink}}
285 {%
286   \ifundefined{aeb@aeb_mlink@opts}{}
287   {\edef\temp@exp{\ifpdfmarkup\noexpand\ap@RP{aeb_mlink}\else
288     \noexpand\aebp@PkgWarningii{aeb_mlink}\fi}\temp@exp}
289 }
290 \let\execjs\aebp@YES
    (2021/06/20) Move the JavaScript file aeb_pro.js to the acrotex-js package, dated
    2021/06/19 or later.
291 \IfFileExists{acrotex-js.sty}{\let\reqpkg\relax}
    If the style file acrotex-js.sty is detected, we assume the document author
    has correctly installed aeb_pro.js, if not, we require this package, which may be
    automatically installed on some TEX systems.
292 {\PackageWarningNoLine{aeb_pro}{%
293   The acrotex-js package is required.\MessageBreak
294   Before continuing, install this package,\MessageBreak
295   read the documentation, and place\MessageBreak
296   aeb_pro.js in the expected folder
297   }\def\reqpkg{\usepackage{acrotex-js}[2021/06/19]}
298 }
299 \reqpkg

300 \RequirePackage{aeb-comment}
301 \RequirePackage{calc}
302 \RequirePackage{eso-pic}

    Input forms16be if linktoattachments is taken.
303 \ifx\aeb@InputUnicode@empty\else\expandafter
304 \ap@RequireUnicode\fi
305 \ifpdfmarkup\else\if\aebp@uselayers1
306   \excludecomment{printRollover}%
307   \excludecomment{rollover}%
308 \fi\fi

    The following write command is used by both the layers stuff, and the attachments
    routines.
309 \def\aeb@IWV0{\immediate\write\verbatim@out}

```

## 5 Support for PDF help

`\pdfHelp` The `\pdfHelp` macro requires `eforms`. We define the necessary commands for PDF help, as well as some comments needed for `\texHelp`, which uses layers.

```

310 \newsavebox{\aebsavehelp}
311 \definecolor{webblue}{rgb}{0,0,.8}
312 \newcommand{\pdfHelpIndicatorColor}{webblue}
313 \newcommand{\pdfHelpIndicator}[1]{#1\thinspace$}^{\mbox
314   {\footnotesize\negthinspace\ding{109}}}%
315   \raisebox{.225ex}{\llap{\color{\pdfHelpIndicatorColor}%
316     \tiny?\hspace{1.3ex}}}}$}
317 \def\pdfHelpCnt{0}
318 \newcommand{\pdfHelp}{\@ifstar{\count0=\pdfHelpCnt\relax
319 \advance\count0by1\relax\xdef\pdfHelpCnt{\the\count0 }}%
320   \pdfHelpi{pdfhelp-\pdfHelpCnt}}{\pdfHelpi}}
321 \newcommand{\pdfHelpi}[3]{%
322   \let\rollormargstring\texorpdfstring
323   \sbox{\aebsavehelp}{\pdfHelpIndicator{#3}}%
324   \makebox[Opt][l]{\makespecialJS\pushButton[{\S}{\BG}{\BC}{\H{N}}
325     \TU{#2}]{#1}{\wd\aebsavehelp}{11bp}}\usebox{\aebsavehelp}%
326   \let\rollormargstring@gobbletwo
327 }
328 \newcommand\pdfPHPProof[3]{\small\raggedright\texttt{#3}: #2}
329 \newcommand\pdfPrintHelp{\@ifstar{\count0=\pdfHelpCnt\relax
330 \advance\count0by1\relax\xdef\pdfHelpCnt{\the\count0}}%
331   \pdfPrintHelpi{pdfhelp-\pdfHelpCnt}}{\pdfPrintHelpi}}
332 \newcommand\pdfPrintHelpi[3]{%
333   \let\rollormargstring\texorpdfstring
334   \ifeqforpaper
335     \ifeq@proofing
336       \pdfHelp{#1}{#2}{#3}\marginpar{\Hy@pdfstringfalse
337         \pdfPHPProof{#1}{#2}{\Hy@pdfstringtrue#3}}\else
338         \pdfHelpIndicator{#3}\fi
339   \else
340     \pdfHelp{#1}{#2}{#3}\ifeq@proofing
341     \marginpar{\Hy@pdfstringfalse
342       \pdfPHPProof{#1}{#2}{\Hy@pdfstringtrue#3}}\fi
343   \fi
344   \let\rollormargstring@gobbletwo
345 }
346 % End of package segment
347 </package>
348 <*aeblayers>

```

## 6 Support for OCG

We place PostScript code in the document that supports OCG.

### 6.1 Initialization

```

349 \newif\ifaeb@ocOn \aeb@ocOntrue
350 \ifaeb@ocOn

```

```

351 \AtBeginDocument{%
352 \headerps@out{%
353 {/OCProcSet /ProcSet findresource} stopped not
354 {/initialize get exec}
355 {
356 /BeginOC /pop load def
357 /EndOC {} def
358 /SimpleOC /pop load def
359 /SetOCGInitState {pop pop} bind def
360 /OCEndPage {} def
361 /SetOCGIntent {pop pop} bind def
362 /SetOCGUsage {pop pop} bind def
363 /AddASEvent {pop pop pop} bind def
364 }
365 ifelse
366 }
367 }\fi
368 \ifaeb@oc0n
369   \AtEndDocument{\literalps@out{{/OCProcSet /ProcSet findresource
370   /terminate get exec} stopped pop}}
371 \fi

```

## 6.2 Supporting commands for OCG

In this section we install the necessary code creating and navigating through PDF layers.

### 6.2.1 Document JavaScript in support of OCG

```

372 \begin{insDLJS}[getxBld]{aebocg}{AeB Pro: OCG Support}
373 var ocgs = this.getOCGs();
374 var xBlds = new Array();
375 %console.println("Reindexing ocgs and creating xBlds");
376 if ( ocgs != null ) {
377   var l = ocgs.length;
378   for ( i=0; i<l; i++ ) {
379     if (ocgs[0].name.substring(0,3) == "xb." )
380       xBlds.push(ocgs.shift());
381     else ocgs.push(ocgs.shift());
382   }
383 }
384 var assocxLayers = new Array();
385 if ( xBlds != null ) {
386   for ( var i=0; i<xBlds.length; i++)
387     assocxLayers[xBlds[i].name]=i;
388 }
389 function getxBld (name) {
390   var i = assocxLayers["xb."+name];
391   return ( i == undefined ) ? null : xBlds[i];
392 }

```

```

393 function toggleHelp (name)
394 {
395     var o = getxBld(name);
396     if ( o != null ) o.state = (event.name.indexOf("Enter") != -1);
397 }
398 function toggleSetThisLayer(name, bState) {
399     var oLayer = getxBld(name);
400     if ( oLayer != null )
401         oLayer.state = (arguments.length > 1) ? bState : !oLayer.state;
402 }
403 function getLayerState(name) {
404     var oLayer = getxBld(name);
405     return ( oLayer != null ) ? oLayer.state : null;
406 }
407 \end{insDLJS}

```

## 6.2.2 Macros for creating rollovers

The following commands were taken from the APB package, and were slightly modified.

```

408 \newcommand{\texHelpIndicatorColor}{webblue}
409 \newcommand{\texHelpIndicator}[1]{#1\thinspace$}^{\mbox
410   {\footnotesize\negthinspace\ding{109}}\raisebox{.225ex}
411   {\llap{\color{\texHelpIndicatorColor}%
412     \tiny?\hspace{1.3ex}}}}$}
413 \newcommand{\addJStexHelpEnter}[1]{\def\aebp@addJStexHelpEnter{\r#1}}
414 \newcommand{\addJStexHelpExit}[1]{\def\aebp@addJStexHelpExit{\r#1}}
415 \newcommand{\resetaddJStexHelp}{%
416   \let\aebp@addJStexHelpEnter\@empty
417   \let\aebp@addJStexHelpExit\@empty
418 }
419 \resetaddJStexHelp
420 \newif\iftexhelptoggleOff \texhelptoggleOfftrue

```

Added a star option to `\texHelp`, when present the layer is not closed on exit from the target word.

```

421 \newcommand{\texHelp}{\@ifstar{\texhelptoggleOfffalse\texHelp@i}
422   {\texhelptoggleOfftrue\texHelp@i}}
423 \newcommand{\texHelp@i}[2]{%
424   \let\rollormargstring\texorpdfstring
425   \sbox{\aebsavehelp}{\texHelpIndicator{#2}}%
426   \makebox[0pt][l]{\pushButton[\S{\S}\BG{} \BC{} \H{N}
427     \AA{\AAMouseEnter{\JS{toggleHelp("#1");\aebp@addJStexHelpEnter}}}%
428     \AAMouseExit{\JS{\iftexhelptoggleOff toggleHelp("#1");\fi
429       \aebp@addJStexHelpExit}}}] {#1}%

```

Changed height calculation in `\texHelp` to allow a rollover of more than text.

```

430   {\wd\aebsavehelp}{\ht\aebsavehelp+\dp\aebsavehelp}}%
431   \usebox{\aebsavehelp}\resetaddJStexHelp
432   \let\rollormargstring\@gobbletwo
433 }

```

### 6.2.3 Macros for layers

We have some basic commands for creating layers in AeB Pro.

```

434 \newif\ifaeb@SimpleOCOn \aeb@SimpleOCOnfalse
435 \newcommand{\@SOC}[2][false]{\global\ifaeb@SimpleOCOntrue%
436   \ifaeb@ocOn\literalps@out{(#2) SimpleOC (#2) #1 SetOCGInitState}\fi}
437 \newcommand{\b@OC}[2][false]{\ifaeb@SimpleOCOn\@aebSimpleNesterror\fi
438   \ifaeb@ocOn\literalps@out{(#2) BeginOC (#2) #1 SetOCGInitState}\fi}
439 \newcommand{\e@OC}{\ifaeb@ocOn\literalps@out{EndOC}\fi}

```

`\xBld` We have some basic commands for creating layers in AeB Pro. The syntax for `\xBld` is as follows:

```

\xBld[true|false|print=(true|false)]
  {<layer_name><content in layer>\eBld

```

The description of the optional parameters follows:

- true**
  - **true**: Inserting **true** as an optional argument causes the layer to be initially visible.
- false**
  - **false**: Inserting **false** as an optional argument causes the layer to be initially hidden. The default is **false**, so if no **true** or **false** argument is passed as an optional argument, the layer will be initially hidden.
- print**
  - **print**: This key allows you to set the printability of the layer: If **print=true**, the layer *always prints*, no matter if it is visible or not. If **print=false**, the layer *never prints*, no matter if it is visible or not. If the **print** key is not specified, then the layer will print if visible, otherwise, it does not print. Normally, the **print** key is not specified, and the layer is printed if visible.

```

440 \newcommand{\xBld}[2][]{%
441   \setkeys{aebpxBld}{false,#1}\b@OC[\aebpxBld@visible]{xb.#2}%
442   \ifx\aebpxBld@print\@empty\else
443     \literalps@out{/Print [/Print] [(xb.#2)] AddASEvent}%
444     \ifAEBP@aebpxBld@print
445       \literalps@out{(xb.#2) << /Print<<
446         /PrintState/ON >> >> SetOCGUsage}%
447     \else
448       \literalps@out{(xb.#2) << /Print<<
449         /PrintState/OFF >> >> SetOCGUsage}%
450     \fi
451     \global\let\aebpxBld@print\@empty
452   \fi
453 }
454 \let\eBld\e@OC
455 \def\@aebSimpleNesterror{\PackageWarning{aeb}
456 {The scope of the last \string\Bld\space command
457 \MessageBreak includes the scope of the nested Bld.
458 \MessageBreak Executing \protect\ocOff\space to correct this.
459 \MessageBreak The problem occurs}\ocOff}

```



The keyval definitions of the optional parameters for `\xBld`. (2015/08/08) Made `print` key into a choice field.

```

460 \newif\ifAEBP@aebpxBld@print \AEBP@aebpxBld@printfalse
461 \define@choicekey+{aebpxBld}{print}[\val\nr]{true,false,%
462 visible,always,never}[visible]{%
463 \ifcase\nr\relax
464 \def\aebpxBld@print{true}\AEBP@aebpxBld@printtrue\or
465 \def\aebpxBld@print{false}\AEBP@aebpxBld@printfalse\or
466 \let\aebpxBld@print\@empty\or
467 \def\aebpxBld@print{true}\AEBP@aebpxBld@printtrue\or
468 \def\aebpxBld@print{false}\AEBP@aebpxBld@printfalse\fi
469 }{\PackageWarning{aeb_pro}{Bad choice for initState, permissible values
470 are true, false, visible, always, and never. Try again}}
471 \let\aebpxBld@print\@empty
472 \define@key{aebpxBld}{true}[true]{\def\aebpxBld@visible{#1}}
473 \define@key{aebpxBld}{false}[false]{\def\aebpxBld@visible{#1}}

```

`initState` The `initState` sets the initial state of the OCG, the default is `false`, meaning the layer is initially not shown.

```

474 \define@choicekey+{aebpxBld}{initState}[\val\nr]{on,off,%
475 true,false}[false]{%
476 \ifcase\nr\relax
477 \def\aebpxBld@visible{true}\or
478 \def\aebpxBld@visible{false}\or
479 \def\aebpxBld@visible{true}\or
480 \def\aebpxBld@visible{false}\fi
481 }{\PackageWarning{aeb_pro}{Bad choice for initState, permissible values
482 are on, off, true and false. Try again}}

```

#### 6.2.4 The behavior of the rollovers

`\defineRC` RC stands for rollover content. `\defineRC` is used to define the content of the rollover so the content can be used for the `rollover` and the `printRollover` environments. These two environments are comment environments that are included or excluded depending on various conditions.

The command `\csarg` is a convenience command defined in the `comment` package. The `comment` package is required so this command is available.

```

483 \long\def\defineRC#1#2{\csarg\def{RC#1}{#2}}
484 \def\insertRC#1{\@nameuse{RC#1}}
485 \long\def\definePR#1#2{%
486 \ifeqforpaper\csarg\def{PR#1}{#2}\else
487 \ifeq@proofing\csarg\def{PR#1}{#2}\else
488 \csarg\let{PR#1}\relax\fi\fi
489 }
490 \def\insertPR#1{%
491 \ifeq@proofing\@nameuse{PR#1}\else
492 \csarg\let{insertPR}\@gobble\fi
493 \csarg\let{PR#1}\relax
494 }

```

```

    rollover    Depending on the switches \ifeqforpaper and \ifeq@proofing, the comment
printRollover environments rollover and printRollover are included or excluded. Also de-
\definePR      fined here are the two commands \definePR and \insertPR (PR stands for print
\insertPR      rollover). These two commands are set equal to either \defineRC and \insertRC,
               or \relax, depending on the switches.
               If exerquiz is not loaded, then the switch \ifeq@proofing is undefined. We
               must define this switch in this case, and set it to false.
495 \@ifundefined{ifeqforpaper}{\newif\ifeqforpaper \eqforpaperfalse}{}
496 \@ifundefined{ifeq@proofing}{\newif\ifeq@proofing \eq@proofingfalse}{}
497 \providecommand\turnProofingOn{\eq@proofingtrue}
498 \providecommand\turnProofingOff{\eq@proofingfalse}
\proofRollovers turns on the printRollover environment, if there any defined in the document.
499 \def\proofRollovers{%
500     \ifeqforpaper
501         \ifeq@proofing
502             \includecomment{rollover}%
503             \includecomment{printRollover}%
504         \else
505             \excludecomment{printRollover}%
506             \includecomment{rollover}%
507         \fi
508     \else
509         \ifeq@proofing
510             \includecomment{rollover}%
511             \includecomment{printRollover}%
512         \else
513             \excludecomment{printRollover}%
514             \includecomment{rollover}%
515         \fi
516     \fi
517 }
               We expand \proofRollovers immediately, but may be expanded in preamble
               with \turnProofingOn.
518 \proofRollovers
519 \@onlypreamble\proofRollovers

\textPrintHelp The \textPrintHelp command reduces to \texHelp or typesets its second argu-
               ment.
520 \newcommand\texPHProof[2]{\small\raggedright\texttt{#1}: #2}
521 \let\rollormargstring@gobbletwo
522 \def\textPrintHelp#1#2{%
523     \let\rollormargstring\texorpdfstring
524     \ifeqforpaper
525         \ifeq@proofing
526             \texHelp{#1}{#2}\marginpar{\Hy@pdfstringfalse
527                 \texPHProof{#1}{#2}}\else
528                 \texHelpIndicator{#2}\fi

```

```

529   \else
530       \texHelp{#1}{#2}\ifeq@proofing\marginpar{\Hy@pdfstringfalse
531           \texPHProof{#1}{#2}}\fi
532   \fi
533   \let\rollormargstring@gobbletwo
534 }

```

## 6.3 In Support of Anime using Layers

Using layers, it is possible to create a sort of animation (anime). The following are some rudimentary methods for creating anime. An example can be found below, as well as in `aebpro_ex4.tex`.

### 6.3.1 L<sup>A</sup>T<sub>E</sub>X Support

`\animeBld` The command `\animeBld` increments `ocSeq`, and names the layer so that the JS `\DeclareAnime` can run through the layers in an orderly way.

`\DeclareAnime` takes three arguments, the anime base name, the speed at which you wish the anime to be played, and the number of frames of this animation. It saves these three parameters in macros, and creates `\animBldName`, which is used in the creation of the anime. See example file `aebpro_ex4.tex`.

```

535 \newcommand{\animeBld}[1][\stepcounter{ocSeq}]%
536     \xBld[#1]{\animBldName}
537 \newcounter{ocSeq}

```

(10/01/01) We give an animation the special ‘anime!!’ prefix, so JavaScript can identify animation layers.

```

538 \newcommand{\animeName}[1]{anime!!#1}%01/01/2010
539 \newcommand{\DeclareAnime}[3]
540 {%
541     \setcounter{ocSeq}{0}%
542     \xdef\animBaseName{anime!!#1}%01/01/2010
543     \xdef\animSpeed{#2}\xdef\animFrames{#3}%
544     \gdef\animBldName{\animBaseName#\theocSeq}%
545 }

```

`\backAnimeBtn` The command `\animeBld` increments `ocSeq`, and names the layer so that the JS `\clearAnimeBtn` can run through the layers in an orderly way.

`\forwardAnimeBtn` `\DeclareAnime` takes three arguments, the anime base name, the speed at which you wish the anime to be played, and the number of frames of this animation. It saves these three parameters in macros, and creates `\animBldName`, which is used in the creation of the anime. See example file `aebpro_ex4.tex`.

- #1: This is an optional parameter for changing the appearance of the button, See the `efrms` package documentation.
- #2: width of the button
- #3: height of the button

```

546 \newcommand{\backAnimeBtn}[3] []
547 {%
548   \pushButton[\textFont{Wingdings3}\uCA{FEFF00C5}\TU{Reverse}#1
549   \A{\JS{%
550     try { aebAnimeTimeOutF.running=false; } catch(e) {} \r
551     try { aebAnimeTimeOutB.running=false; } catch(e) {} \r
552     aebAnimeLayersBackward(\animSpeed,\nFrames,"\animBaseName");
553   }]{\animBaseName BtnBack}{#2}{#3}%
554 }
555 \newcommand{\clearAnimeBtn}[3] []
556 {%
557   \pushButton[\textFont{Wingdings}\CA{n}\TU{Clear}#1
558   \A{\JS{%
559     try { aebAnimeTimeOutF.running=false; } catch(e) {} \r
560     try { aebAnimeTimeOutB.running=false; } catch(e) {} \r
561     aebAnimeLayersClear(\animSpeed,\nFrames,"\animBaseName");
562   }]{\animBaseName BtnClear}{#2}{#3}%
563 }
564 \newcommand{\forwardAnimeBtn}[3] []
565 {%
566   \pushButton[\textFont{Wingdings3}\uCA{FEFF00C6}\TU{Graph It!}#1
567   \A{\JS{%
568     try { aebAnimeTimeOutF.running=false; } catch(e) {} \r
569     try { aebAnimeTimeOutB.running=false; } catch(e) {} \r
570     aebAnimeLayersForward(\animSpeed,\nFrames,"\animBaseName");
571   }]{\animBaseName BtnForward}{#2}{#3}%
572 }

```

Here is a simple example if anime, using the above commands. The code below uses the pstricks-add and fp packages.

```

\begin{center}
\DeclareAnime{sinegraph}{10}{40}
\def\thisframe{\animeBld\psplot[linecolor=red]{0}{\xi}{\sin(x)}\eBld}
\psset{llx=-12pt, lly=-12pt, urx=12pt, ury=12pt}
\begin{psgraph*}[arrows=->] (0,0) (-.5,-1.5) (6.5,1.5){164pt}{70pt}
  \psset{algebraic=true}%
  \rput(4,1){\mathit{\$y=\sin(x)\$}}
  \FPdiv{\myDelta}{\psPiTwo}{\nFrames}%
  \def\xi{0}%
  \multido{\i=1+1}{\nFrames}{\FPadd{\xi}{\xi}{\myDelta}\thisframe}
\end{psgraph*}

\backAnimeBtn{24bp}{12bp}\kern1bp\clearAnimeBtn{24bp}{12bp}\kern1bp
\forwardAnimeBtn{24bp}{12bp}
\end{center}

```

### 6.3.2 JavaScript Support

```

573 \begin{insDLJS}[_animeLayers]{animelayers}

```

```

574     {AeB Pro: JS in Support of Anime}
575 var _animeLayers = true;
576 var aebAnimeTimeOutF = new Object();
577 aebAnimeTimeOutF.running=false;
578 var aebAnimeTimeOutB = new Object();
579 aebAnimeTimeOutB.running=false;
580 var aebAnimeTimeOutC = new Object();
581 function aebAnimeLayersForward(animSpeed,nFrames,baseName)
582 {
583     var timeinterval = animSpeed;
584     aebAnimeTimeOutF = app.setInterval("aebAnimeTimeOutF.count++;"
585         + "if (aebAnimeTimeOutF.count > "
586         + nFrames + " || !aebAnimeTimeOutF.running )"
587         + " app.clearInterval(aebAnimeTimeOutF);\r"
588         + "else { if (aebAnimeTimeOutF.count == 1 )"
589         + " getxBld('"+baseName+"\#1').state=true;\t"
590         + "else { getxBld('"+baseName+"\#"
591         + "(aebAnimeTimeOutF.count-1)).state=false;\t"
592         + "getxBld('"+baseName+"\#"
593         + "+aebAnimeTimeOutF.count).state=true;}}", timeinterval);
594     aebAnimeTimeOutF.running = true;
595     aebAnimeTimeOutF.start = 1;
596     aebAnimeTimeOutF.count=0;
597 }
598 function aebAnimeLayersBackward(animSpeed,nFrames,baseName)
599 {
600     var timeinterval = animSpeed;
601     aebAnimeTimeOutB = app.setInterval("aebAnimeTimeOutB.count++;\r"
602         + "if (aebAnimeTimeOutB.count > "+nFrames
603         + "|| !aebAnimeTimeOutB.running )"
604         + " app.clearInterval(aebAnimeTimeOutB);\r"
605         + "else { if (aebAnimeTimeOutB.count == "
606         + nFrames+") getxBld('"+baseName+"\#1').state=false;\r"
607         + "else { getxBld('"+baseName+"\#"
608         + "+(aebAnimeTimeOutB.start-%
609 aebAnimeTimeOutB.count+1)).state=false;\r"
610         + "getxBld('"+baseName+"\#"
611         + "+(aebAnimeTimeOutB.start-%
612 aebAnimeTimeOutB.count)).state=true;}}",
613         timeinterval);
614     aebAnimeTimeOutB.running = true;
615     aebAnimeTimeOutB.start = nFrames;
616     aebAnimeTimeOutB.count=0;
617 }
618 function aebAnimeLayersClear(animSpeed,nFrames,baseName)
619 {
620     aebAnimeTimeOutF.running = false;
621     aebAnimeTimeOutB.running = false;
622     clearFrames = app.setTimeout(
623         "hideAllFrames("+nFrames+",\""+baseName+"\")",animSpeed);

```

```

624     var timeinterval = animSpeed;
625     aebAnimeTimeOutC = app.setInterval("aebAnimeTimeOutC.count++;r"
626     + "if (aebAnimeTimeOutC.count >= "+nFrames
627     +") app.clearInterval(aebAnimeTimeOutC);r"
628     + "else { getxBld('"+baseName+"\#' "
629     +"(aebAnimeTimeOutC.start-aebAnimeTimeOutC.count)).state=false;}",
630     timeinterval);
631     aebAnimeTimeOutC.start = nFrames;
632     aebAnimeTimeOutC.count=0;
633 }
634 function hideAllFrames(n, baseName)
635 {
636     for ( var i=1; i <= n ; i++ )
637         getxBld(baseName+"\#" +i).state=false;
638 }
639 \end{insDLJS}

640 % End of the layers segment
641 \</aeblayers>
642 \<*aebpro>

```

## 7 Initial View

Supported preferences include HideToolbar, HideMenubar, FitWindow, Center-Window, DisplayDocTitle, NonFullScreenPageMode (UseNone, UseOutlines, UseThumbs, UseOC, UseAttachments)

### 7.1 Window Options

```

fit      The keys recognized for the windows options are fit, center, fullscreen, and
center  showtitle.
fullscreen 643 \@tfor\ aeb@windowopts={fit}{FitWindow}{center}{CenterWindow}}%
showtitle 644     {{fullscreen}{FullScreen}}{{showtitle}{DisplayDocTitle}}\do
645 {%
646     \expandafter\getargsii\ aeb@windowopts
647     \edef\temp@expand@def{%
648         \noexpand\define@key{aebWindowOpts}{\ aeb@argi}[true]{%
649         \noexpand\def\csname aeb@windowopts\ aeb@argii\endcsname{%
650             /\ aeb@argii\space##1}%
651         }%
652     }\temp@expand@def
653     \csarg\let{aeb@windowopts\ aeb@argii}\@empty
654 }

\setWindowOptions Set the User Window options on the Initial View tab of the Document Properties
                  dialog box.
655 \newcommand{\setWindowOptions}[1]{\cckvsetkeys{aebWindowOpts}{#1}}

```

## 7.2 User Interface Options

`hidetoolbar` Key recognized for the use interface options are `hidetoolbar`, `hidemenubar`, and  
`hidemenubar` `hidewindowui`.  
`hidewindowui`

```
656 \@tfor\@eb@uiopts:={{hidetoolbar}{HideToolbar}}
657   {{hidemenubar}{HideMenubar}}{{hidewindowui}{HideWindowUI}}\do{%
658   \expandafter\getargsii\@eb@uiopts
659   \edef\temp@expand@def{%
660     \noexpand\define@key{\@ebUIOpts}{\@eb@argi}[true]%
661     {%
662       \noexpand\edef\csname \@eb@uiopts\@eb@argii\endcsname{%
663         /\@eb@argii\space##1}%
664     }%
665   }\temp@expand@def
666   \csarg\let{\@eb@uiopts\@eb@argii}\@empty
667 }
```

`\setUIOptions` Set the User Interface options on the Initial View tab of the Document Properties dialog box.

```
668 \newcommand{\setUIOptions}[1]{\cxkvsetkeys{\@ebUIOpts}{#1}}
```

## 7.3 Layout and Magnification

### 7.3.1 Navigation Tab (Page Mode)

`navitab` Supported values for the `navitab` key are:

- `UseNone`: Page Only
- `UseOutlines`: Bookmarks Panel and Page
- `UseThumbs`: Pages Panel and Page
- `FullScreen`: Fullscreen mode
- `UseOC`: Layers Panel and Page
- `UseAttachments`: Attachments Panel and Page

```
669 \define@choicekey+{\@ebLayoutMag}{navitab}{UseNone,UseOutlines,%
670   UseThumbs,FullScreen,UseOC,UseAttachments}[UseNone]{%
671   \def\@eb@layoutMagNaviTab{#1}%
672 }\PackageWarning{\@eb}{Bad choice for PageLayout, permissible values
673   are UseNone, UseOutlines, UseThumbs, FullScreen,%
674   UseOC, UseAttachments. Try again}}
675 \let\@eb@layoutMagNaviTab\@empty
```

### 7.3.2 Page Layout

`pagelayout` Supported value for the `pagelayout` key are:

- `SinglePage`: Single page
- `OneColumn`: Single page continuous
- `TwoPageLeft`: Two-Up (Facing)
- `TwoColumnLeft`: Two-Up Continuous (Facing)
- `TwoPageRight`: Two-Up (Cover Page)
- `TwoColumnRight`: Two-Up Continuous (Cover Page)

```
676 \define@choicekey+{aebLayoutMag}{pagelayout}{SinglePage,OneColumn,%
677     TwoColumnLeft,TwoColumnRight,TwoPageLeft,TwoPageRight}[SinglePage]{%
678     \def\aeb@layoutMagPageLayout{#1}%
679 }{\PackageWarning{aeb}{Bad choice for PageLayout, permissible values
680  are SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight,
681  TwoPageLeft,TwoPageRight. Try again}}
682 \let\aeb@layoutMagPageLayout\@empty
```

### 7.3.3 Magnification

`mag` Supported values for the `mag` key are:

```
Actual Size <</D[1 0 R/XYZ -32768 -32768 1.0]/S/GoTo>>
Fit Page <</D[1 0 R/Fit]/S/GoTo>>
Fit Width <</D[21 0 R/FitH -32768]/S/GoTo>>
Fit Height <</D[1 0 R/FitV -32768]/S/GoTo>>
Fit Visible <</D[21 0 R/FitBH -32768]/S/GoTo>>
```

```
683 \define@choicekey+{aebLayoutMag}{mag}{\val\nr}%
684     {ActualSize,FitPage,FitWidth,FitHeight,FitVisible}[FitPage]%
685 {%
686     \ifcase\nr\relax
687         \edef\aeb@layoutMagMag{\dl@actualsize}\or
688         \def\aeb@layoutMagMag{\dl@fitpage}\or
689         \def\aeb@layoutMagMag{\dl@fitwidth}\or
690         \def\aeb@layoutMagMag{\dl@fitheight}\or
691         \def\aeb@layoutMagMag{\dl@fitvisible}\fi
```

Change division from postscript operation to a tex calculation to support the `useacrobat` option.

```
692 }{\setlength\@tempdima{1pt*\ratio{\val pt}{100pt}}}%
693 \edef\aeb@layoutMagMag{\dl@mag{\strip@pt\@tempdima}}
694 \let\aeb@layoutMagMag\@empty
```



### 7.3.4 Open to page

`openatpage` The `openatpage` key takes a page number as its value.

```
695 \define@key{aebLayoutMag}{openatpage}[1]{\def\ae@openatpageChoice{#1}%  
696   \ifpdfmarkup\def\ae@layoutMagPage{Page#1}\else  
697   \def\ae@layoutMagPage{#1}\fi}  
698 \let\ae@openatpageChoice\@empty  
699 \let\ae@layoutMagPage\@empty
```

`\setLayoutMag` Set the Layout and Magnifications options on the Initial View tab of the Document Properties dialog box.

```
700 \newcommand{\setLayoutMag}[1]{\setkeys{aebLayoutMag}{openatpage=1}%  
701   \cckvsetkeys{aebLayoutMag}{#1}}
```

`\DeclareInitView` This command is the ultimate holder of the Initial View data. It has three keys `layoutmag`, `windowoptions` and `uioptions`. The values for each of these three are key values for the commands defined above: `\setLayoutMag`, `\setWindowOptions` and `\setUIOptions`.

```
702 \define@key{aebDeclareInitView}{layoutmag}  
703   {\def\aeDeclareInitView@layoutmag{#1}}  
704 \let\aeDeclareInitView@layoutmag\@empty  
705 \define@key{aebDeclareInitView}{windowoptions}  
706   {\def\aeDeclareInitView@windowoptions{#1}}  
707 \let\aeDeclareInitView@windowoptions\@empty  
708 \define@key{aebDeclareInitView}{uioptions}  
709   {\def\aeDeclareInitView@uioptions{#1}}  
710 \let\aeDeclareInitView@uioptions\@empty  
711 \newcommand{\DeclareInitView}[1]{%  
712   \setkeys{aebDeclareInitView}{#1}%  
713   \edef\temp@expand@sets{%  
714     \noexpand\setLayoutMag{\aeDeclareInitView@layoutmag}%  
715     \noexpand\setWindowOptions{\aeDeclareInitView@windowoptions}%  
716     \noexpand\setUIOptions{\aeDeclareInitView@uioptions}%  
717   }\temp@expand@sets  
718 }
```

`\taggedPDF` Added 07/08/09 to support tagged pdf, principally tabbing using structure. Strictly speaking, the document need not be marked as tagged.

```
719 \providecommand{\taggedPDF}{\def\ae@TaggedPDF{true}}  
720 \let\ae@TaggedPDF\@empty  
721 \@onlypreamble\taggedPDF
```

Additional Open action

`\additionalOpenAction` Document open action, this command is renamed to `\addToDocOpen` and enhanced.

`\addToDocOpen` This command enhances `\additionalOpenAction`. This version can be used several times in the preamble with possibly different action types. Use the command

`\addToDocOpen` to include additional action to be performed when the document opens. This is different from the actions that may be performed when the first page is opened. In fact, *this action is independent of what what page is initially opened.*

`\requiresVersion` `\requiresVersion` can be used to put a version condition on viewing the document. `\requiresVersion` uses `\addToDocOpen`. If you want to combine several actions, including a version requirement you'll have to use `\addToDocOpen`. Usage:

```
\requiresVersion{7}
```

`\alternateDocumentURL` An URL to direct the user to, it is used by `\requiresVersion` when the version requirement is not met.

`\requiredVersionMsg` `\requireVersionMsg` is the message that is displayed when the version condition is not met.

`\requiredVersionMsgRedirect` An redirect message when the version requirement fails, and the document is in a browser.

`\afterRequirementPassedJS` A hook into the end of the `\requiresVersion` command for additional JS to be executed.

```
722 \let\aeB@addOpenAction\@empty
723 \let\aeB@addActionCode\@empty
724 \let\aeB@addOpenActionArg\@empty
725 \def\AEB@psMrk{[\space}
726 \let\aeB@popentoks\@empty
727 \let\aeB@FA\aeB@NO
728 \def\aeB@rightDelimiters{}
```

`\addToDocOpen` The argument of `\addToDocOpen` is an action, usually JavaScript or named.

```
\addToDocOpen{/S/GoTo/D[\Page{2}/Fit]}
\addToDocOpen{\JS{console.println("Public: Open it baby!");}}
```

As of 2015/06/15, the original version of `\addToDocOpen` was moved to `insdljs` (v2.1). This redefinition extends that capability slightly by incorporating into it the `\dev@addToDocOpen` command, see definition below.

The `\addToDocOpen` command is the major interface between the document author for inserting open actions (not first page actions). At least in the case of `pdflatex`, `hyperref` has the potential of overwriting the user's choices with its own choice. We turn off the open page action of `hyperref` by setting the following two commands to empty.

```
729 \let\@pdfstartview\@empty
730 \let\@pdfstartpage\@empty
```

Now for the definition of `\addToDocOpen`.

```
731 \renewcommand{\addToDocOpen}[1]{%
```

The `\aebpFA` flag indicates whether this command has been used. The first it is used, we set things up.

```

732   \ifx\aebpFA\aebp@NO
733     \gdef\aeb@addOpenActionArg{#1}%
734     \g@addto@macro\aebpopentoks{#1}%
735     \def\aeb@addActionCode{\bgroup\makeJSSpecials\d\l@preProcDefns
736 \ifpdfmarkup
737   \literalps@out{\AEB@psMrk/_objdef {pubAddToDocOpen}
738     /type/dict /OBJ pdfmark^^J%
739   \AEB@psMrk{pubAddToDocOpen}
740     << \aebpopentoks
741       \ifx\dev@addOpenAction\@empty\else
742         \ifx\aeb@addOpenActionArg\@empty\else^^J%
743         \dev@addOpenAction\fi\fi\aebp@rightDelimiters
744       >> /PUT pdfmark}%
745   \gdef\pubAddToDocOpen{{pubAddToDocOpen}}%
746   \xdef\aeb@addOpenAction{/Next \pubAddToDocOpen}%
747 \else\ifpdf
748   \immediate\pdfobj{<< \aebpopentoks
749     \ifx\dev@addOpenAction\@empty\else
750     \ifx\aeb@addOpenActionArg\@empty\else^^J%
751     \dev@addOpenAction\fi\fi\aebp@rightDelimiters
752     >>}\xdef\pubAddToDocOpen{\the\pdflastobj\space 0 R}%
753   \xdef\aeb@addOpenAction{/Next \pubAddToDocOpen}%
754 \else\ifxetex
755   \immediate\@pdfm@mark{obj @pubAddToDocOpen << \aebpopentoks
756     \ifx\dev@addOpenAction\@empty\else
757     \ifx\aeb@addOpenActionArg\@empty\else^^J%
758     \dev@addOpenAction\fi\fi\aebp@rightDelimiters >> }%
759   \gdef\pubAddToDocOpen{@pubAddToDocOpen}%
760   \xdef\aeb@addOpenAction{/Next \pubAddToDocOpen}%
761 \fi\fi\fi\egroup}%

```

Set the `\aebpFA` flag to `\aebp@YES` to indicate this command has been executed at least once.

```

762   \global\let\aebpFA\aebp@YES
763   \else

```

If the document author uses this command a second time, the code goes to here.

```

764     \edef\aebp@d\jstmp{\aebp@rightDelimiters}%
765     \xdef\aebp@rightDelimiters{\aebp@d\jstmp >> }%
766     \g@addto@macro\aebpopentoks{/Next << #1 }%
767   \fi
768 }
769 \let\additionalOpenAction\addToDocOpen

```

`\dev@addToDocOpen` The `\dev@addToDocOpen` is similar to `\addToDocOpen` but is meant for package developers. These actions are executed *after the actions* of `\addToDocOpen`.

```

770 \let\dev@addOpenAction\@empty
771 \let\dev@addOpenActionArg\@empty

```

```

772 \let\dev@addActionCode\@empty
773 \let\aebpFAP\aebp@NO
774 \let\aebp@rightDelimitersP\@empty
775 \def\aebp@rightDelimitersP{}
776 \newcommand{\dev@addDocOpen}[1]{%
    The \aebpFAP flag indicates whether this command has been used. The first it is
    used, we set things up.
777     \ifx\aebpFAP\aebp@NO
778         \gdef\dev@addOpenActionArg{#1}%
779         \g@addto@macro\aebp@rightDelimitersP{#1}%
780         \def\dev@addActionCode{\bgroup\makeJSSpecials\dl@preProcDefs
781 \ifpdfmarkup
782     \literalps@out{\AEB@psMrk/_objdef {privAddDocOpen}
783     /type/dict /OBJ pdfmark^^J%
784     \AEB@psMrk{privAddDocOpen} << \aebp@rightDelimitersP
785     \aebp@rightDelimitersP\space>> /PUT pdfmark}%
786     \gdef\privAddDocOpen{{privAddDocOpen}}%
787     \gdef\dev@addOpenAction{/Next {privAddDocOpen}}%
788 \else\ifpdf
789     \immediate\pdfobj{<< \aebp@rightDelimitersP
790     \aebp@rightDelimitersP\space>>}%
791     \xdef\privAddDocOpen{\the\pdfobj\space 0 R}%
792     \xdef\dev@addOpenAction{/Next \privAddDocOpen}%
793 \else\ifxetex
794     \immediate\@pdfm@mark{obj @privAddToDocOpen << \aebp@rightDelimitersP
795     \aebp@rightDelimitersP\space>> }%
796     \gdef\privAddDocOpen{@privAddToDocOpen}%
797     \gdef\dev@addOpenAction{/Next \privAddDocOpen}%
798 \fi\fi\fi\egroup}%
    Set the \aebpFAP flag to \aebp@YES to indicate this command has been executed
    at least once.
799     \global\let\aebpFAP\aebp@YES
    If the package developer uses this command a second time, the code goes to here.
800     \else
801         \edef\aebp@dljstmpP{\aebp@rightDelimitersP}%
802         \xdef\aebp@rightDelimitersP{\aebp@dljstmpP >> }%
803         \g@addto@macro\aebp@rightDelimitersP{/Next << #1 }%
804     \fi
805 }

806 \def\aebp@warnonly{warnonly}
807 \newcommand{\aebp@app@title}{AcroTeX Message}

    #1 = warnonly or empty
    #2 = version number

808 \newcommand{\requiresVersion}[2] [] {\def\aebp@argi{#1}%
809     \def\requiredVersionNumber{#2}\dev@addDocOpen{\JS{%
```

```

810 \ifx\aebp@argi\aebp@warnonly
811   var warnOnly=true;
812 \else
813   var warnOnly=false;
814 \fi

```

(2015/10/14) Inserted parentheses to prevent dvips from breaking lines within a string.

```

815   var reqVerMsgC("\aebp@requiredVersionMsg\space")
816     + ((warnOnly) ? ("\aebp@reqVerMsgW") : ("\aebp@reqVerMsgC"));
817   var reqVerMsgR("\aebp@requiredVersionMsgRedirect\space")
818     + ((warnOnly) ? ("\aebp@reqVerMsgW") : ("\aebp@reqVerMsgR"));
819   if( app.viewerVersion < #2 ) {
820     var viewerPassedReq=false;
821     try {
822       if (this.external) {
823         app.alert({cTitle:"\aebp@app@title", cMsg: reqVerMsgR});
824         if (!warnOnly) {
825           if (app.viewerVersion >= 7.0)
826             app.launchURL("\aebp@AlternateDocumentURL");
827           else
828             this.getURL("\aebp@AlternateDocumentURL",false);
829         }
830       } else {
831         app.alert({cTitle:"\aebp@app@title", cMsg: reqVerMsgC});
832         if (!warnOnly) this.closeDoc(true);
833       }
834     } catch(e) {}
835   } else {
836     var viewerPassedReq=true;
837     \aebp@ViewerPassedRequirementJS
838   }
839 }}}
840 \def\requiredVersionNumber{7}
841 \newcommand{\alternateDocumentURL}[1]{\def\aebp@AlternateDocumentURL{#1}}
842 \alternateDocumentURL{http://www.acrotex.net/}
843 \newcommand{\requiredVersionMsg}[1]{\def\aebp@requiredVersionMsg{#1}}
844 \requiredVersionMsg{This document requires Adobe Reader or Acrobat,
845   version \requiredVersionNumber\space or later.}
846 \newcommand{\requiredVersionResult}[3]{\def\aebp@reqVerMsgC{#1}%
847   \def\aebp@reqVerMsgR{#2}\def\aebp@reqVerMsgW{#3}}
848 \requiredVersionResult{The document is now closing.}
849   {Redirecting browser to an alternate page.}
850   {Some features of this document may not work properly.}
851 \newcommand{\requiredVersionMsgRedirect}[1]{%
852   \def\aebp@requiredVersionMsgRedirect{#1}}
853 \requiredVersionMsgRedirect{%
854   This document requires Adobe Reader or Acrobat,
855   version \requiredVersionNumber\space or later.}
856 \newcommand{\afterRequirementPassedJS}[1]{%

```

```

857 \def\ae@ViewerPassedRequirementJS{#1}
858 \let\ae@ViewerPassedRequirementJS\@empty

```

Now let's put it all together. The two flags, \ae@pFA and \ae@pFAP indicate whether the commands \ae@addDocOpen and \ae@dev@addDocOpen were used.

```

859 \def\ae@SetCatalog{%
860 \ifx\ae@pFA\ae@pNO\ae@addDocOpen{JS{;}}\fi
861 \ifx\ae@pFAP\ae@pYES\ae@dev@addActionCode\fi
862 \ae@addActionCode

```

The viewer preferences.

```

863 \def\ae@CatEntry{/ViewerPreferences<<%
864 \ifx\ae@windowoptsFitWindow\@empty\else
865 \ae@windowoptsFitWindow\fi
866 \ifx\ae@windowoptsCenterWindow\@empty\else
867 \ae@windowoptsCenterWindow\fi
868 \ifx\ae@windowoptsDisplayDocTitle\@empty\else
869 \ae@windowoptsDisplayDocTitle\fi
870 \ifx\ae@uioptsHideToolbar\@empty\else
871 \ae@uioptsHideToolbar\fi
872 \ifx\ae@uioptsHideMenubar\@empty\else
873 \ae@uioptsHideMenubar\fi
874 \ifx\ae@uioptsHideWindowUI\@empty\else
875 \ae@uioptsHideWindowUI\fi
876 \ifx\ae@layoutMagNaviTab\@empty\else
877 /NonFullScreenPageMode/\ae@layoutMagNaviTab\fi
878 >>%
879 \ifx\ae@TaggedPDF\@empty\else
880 /MarkInfo<</Marked true>>\fi
881 \ifx\ae@layoutMagPageLayout\@empty\else
882 /PageLayout/\ae@layoutMagPageLayout\fi
883 \ifx\ae@windowoptsFullScreen\@empty
884 \ifx\ae@layoutMagNaviTab\@empty\else
885 /PageMode/\ae@layoutMagNaviTab\fi
886 \else/PageMode/FullScreen\fi

```

The open action key (**OpenAction**) is implemented any of three ways, depending on the driver. The command \ae@pubAddToDocOpen is an indirect reference to additional open actions; this is defined in \ae@addDocOpen.

```

887 \ifpdfmarkup
888 \ifx\ae@layoutMagPage\@empty
889 \ifx\ae@layoutMagMag\@empty
890 /OpenAction \ae@pubAddToDocOpen
891 \else/OpenAction<</D[/{Page1}\ae@layoutMagMag]%
892 /S/GoTo\ae@addOpenAction>>\fi
893 \else
894 \ifx\ae@layoutMagMag\@empty
895 /OpenAction<</D[/{\ae@layoutMagPage}/Fit]%
896 /S/GoTo\ae@addOpenAction>>\else
897 /OpenAction<</D[/{\ae@layoutMagPage}\ae@layoutMagMag]%

```

```

898             /S/GoTo\aeB@addOpenAction>>\fi
899     \fi
900 \else\ifpdf
901     \ifx\aeB@layoutMagPage\@empty
902         \ifx\aeB@layoutMagMag\@empty
903             /OpenAction \pubAddToDocOpen
904         \else/OpenAction <<%
905             /D[\pdfpageref\space\space 0 R\aeB@layoutMagMag]%
906             /S/GoTo\aeB@addOpenAction>>\fi
907     \else
908         \ifx\aeB@layoutMagMag\@empty
909             /OpenAction<</D[\expandafter
910                 \pdfpageref\aeB@layoutMagPage\space\space 0 R/Fit]%
911                 /S/GoTo\aeB@addOpenAction>>\else
912             /OpenAction<</D[\expandafter\pdfpageref
913                 \aeB@layoutMagPage\space\space 0 R\aeB@layoutMagMag]%
914                 /S/GoTo\aeB@addOpenAction>>\fi
915     \fi
916 \else\ifxetex
917     \ifx\aeB@layoutMagPage\@empty
918         \ifx\aeB@layoutMagMag\@empty
919             /OpenAction \pubAddToDocOpen
920         \else/OpenAction<</D[@page1 \aeB@layoutMagMag]%
921             /S/GoTo\aeB@addOpenAction>>\fi
922     \else
923         \ifx\aeB@layoutMagMag\@empty
924             /OpenAction<</D[@page\aeB@layoutMagPage\space/Fit]%
925             /S/GoTo\aeB@addOpenAction>>\else
926             /OpenAction<<%
927                 /D[@page\aeB@layoutMagPage\space\aeB@layoutMagMag]%
928                 /S/GoTo\aeB@addOpenAction>>\fi
929     \fi
930 \fi\fi\fi}%
    Load \aeB@SetCatalog only if a Distiller user.
931     \ifpdfmarkup\literalps@out{\AEB@psMrk{Catalog}
932         <<\aeB@CatEntry>> /PUT pdfmark}\else
933     \ifpdf\pdfcatalog{\aeB@CatEntry}\else
934     \ifxetex\immediate\pdfmark{put @catalog <<\aeB@CatEntry>>}%
935     \fi\fi\fi
936 }
937 \AtBeginDocument{\aeB@SetCatalog}

```

## 8 Fullscreen and Transitions

### 8.1 \setDefaultFS

`\setDefaultFS` Use `\setDefaultFS` to set the defaults for fullscreen mode in the Adobe Reader or Acrobat.

```

938 \newcommand{\setDefaultFS}[1]
939 {%
940   \setkeys{aebDefFS}{Trans=Replace}\setkeys{aebDefFS}{#1}%
941   \makeatletter
942   \InputIfFileExists{fsprodef.js}{\PackageWarning{aeb_pro}
943     {Cannot find fsprodef.js, please run aeb_pro.ins again}}
944   \makeatother
945 }
946 \@onlypreamble\setDefaultFS

```

**Trans** Definitions of the values of the Trans key for supported transitions; these are set using the command `\setDefaultFS`.

```

947 \def\aeb@NoTransition{NoTransition}
948 \define@choicekey+{aebDefFS}{Trans}
949 {NoTransition,UncoverLeft,UncoverRight,UncoverDown,UncoverUp,%
950 UncoverLeftDown,UncoverLeftUp,UncoverRightDown,UncoverRightUp,%
951 CoverLeft,CoverRight,CoverDown,CoverUp,CoverLeftDown,CoverLeftUp,%
952 CoverRightDown,CoverRightUp,PushLeft,PushRight,PushDown,PushUp,%
953 PushLeftDown,PushLeftUp,PushRightDown,PushRightUp,FlyInRight,%
954 FlyInLeft,FlyInDown,FlyInUp,FlyOutRight,FlyOutLeft,FlyOutDown,%
955 FlyOutUp,FlyIn,FlyOut,Blend,Fade,Random,Dissolve,%
956 GlitterRight,GlitterDown,GlitterRightDown,BoxIn,BoxOut,%
957 BlindsHorizontal,BlindsVertical,SplitHorizontalIn,%
958 SplitHorizontalOut,SplitVerticalIn,SplitVerticalOut,WipeLeft,%
959 WipeRight,WipeDown,WipeUp,WipeLeftDown,WipeLeftUp,%
960 WipeRightDown,WipeRightUp,Replace,ZoomInDown,ZoomInLeft,%
961 ZoomInLeftDown,ZoomInLeftUp,ZoomInRight,ZoomInRightDown,%
962 ZoomInRightUp,ZoomInUp,ZoomOutDown,ZoomOutLeft,ZoomOutLeftDown,%
963 ZoomOutLeftUp,ZoomOutRight,ZoomOutRightDown,ZoomOutRightUp,%
964 ZoomOutUp,CombHorizontal,CombVertical}[Replace]{%
965   \def\aeb@arg{#1}\ifx\aeb@arg\@empty
966     \let\aeb@fsTran\@empty\else
967     \ifx\aeb@arg\aeb@NoTransition
968       \def\aeb@fsTran{app.fs.defaultTransition = "";\^^J}
969     \else
970       \def\aeb@fsTran{app.fs.defaultTransition = "#1";\^^J}\fi\fi}
971 {\PackageWarning{aeb}{Bad choice, try again. Meanwhile,
972   will use the default, Replace}}%
973   \def\aeb@fsTran{app.fs.defaultTransition = "Replace";\^^J}}

```

**bgColor** The bgColor key.

```

974 \define@key{aebDefFS}{bgColor}[]{%
975   \def\aeb@arg{#1}\ifx\aeb@arg\@empty
976     \let\aeb@fsBGColor\@empty\else
977     \def\aeb@fsBGColor{app.fs.backgroundColor = #1;\^^J}\fi}
978 \let\aeb@fsBGColor\@empty

```

**timeDelay** The timeDelay key.

```

979 \define@key{aebDefFS}{timeDelay}[]{%
980   \def\aeb@arg{#1}\ifx\aeb@arg\@empty

```



```

981 \let\ae@fstimeDelay\@empty\else
982 \def\ae@fstimeDelay{app.fs.timeDelay = #1;^^J}\fi}
983 \let\ae@fstimeDelay\@empty

loop The loop key.
984 \define@boolkey{aeDefFS}{loop}[true]{%
985 \def\ae@fsLoop{app.fs.loop = #1;^^J}}
986 \let\ae@fsLoop\@empty

cursor The cursor key.
987 \define@choicekey+{aeDefFS}{cursor}{hidden,delay,visible}{%
988 \def\ae@arg{#1}\ifx\ae@arg\@empty
989 \let\ae@fscursor\@empty\else
990 \def\ae@fscursor{app.fs.cursor = cursor.#1;^^J}\fi
991 }{\PackageWarning{ae}{Bad choice for cursor, permissible values
992 are hidden, delay and visible. Try again}}
993 \let\ae@fscursor\@empty

escape The escape key.
994 \define@boolkey{aeDefFS}{escape}[true]{%
995 \def\ae@fsEscape{app.fs.escapeExits = #1;^^J}}
996 \let\ae@fsEscape\@empty

clickAdv The clickAdv key.
997 \define@boolkey{aeDefFS}{clickAdv}[true]{%
998 \def\ae@fsclickAdv{app.fs.clickAdvances = #1;^^J}}
999 \let\ae@fsclickAdv\@empty

fullscreen The fullscreen key.
1000 \define@boolkey{aeDefFS}{fullscreen}[true]{%
1001 \def\ae@windowoptsFullScreen{true}%
1002 }
1003 \let\ae@fsFS\@empty

useTimer The useTimer key.
1004 \define@boolkey{aeDefFS}{useTimer}[true]{%
1005 \def\ae@fsuseTimer{app.fs.useTimer = #1;^^J}}
1006 \let\ae@fsuseTimer\@empty

usePageTiming The usePageTiming key.
1007 \define@boolkey{aeDefFS}{usePageTiming}[true]{%
1008 \def\ae@fsusePageTiming{app.fs.usePageTiming = #1;^^J}}
1009 \let\ae@fsusePageTiming\@empty

```

## 8.2 \setPageTransition

**\setPageTransition** By setting the aeTrans family of keys, you can change the transition effect of the next slide.

```

1010 \def\ae@p@PageTransCnt{0}
1011 \def\ae@p@Step@PageTransCnt{\count0=\ae@p@PageTransCnt\relax
1012 \advance\count0by1\relax
1013 \xdef\ae@p@PageTransCnt{\the\count0 }}

```

```

1014 \newcommand{\setPageTransition}[1]{%
1015     \begingroup
1016     \aebp@Step@PageTransCnt
1017     \setkeys{aebTrans}{TransDur,Speed,#1}%
1018     \edef\aeb@pagetranschoices{\aeb@Page@Dur
1019         /Trans <<\aeb@Trans@Dur\aeb@TranChoice>>}%
1020     \csarg{\xdef}{aeb@pagetransCodeL\aebp@PageTransCnt}{%
1021         \ifpdfmarkup
1022             \noexpand\literalps@out{\AEB@psMrk{ThisPage} <<
1023                 \aeb@pagetranschoices\space>> /PUT pdfmark}%
1024         \else\ifpdf

```

There may be open / close page actions already defined, so we combine previous page actions with current page actions. This seems necessary only for pdftex.

```

1025             \noexpand\global\noexpand\pdfpageattr=\noexpand\expandafter
1026             {\noexpand\the\noexpand\pdfpageattr\aeb@pagetranschoices}%
1027         \else\ifxetex
1028             \noexpand\immediate\noexpand\@pdfm@mark{put @thispage <<
1029                 \aeb@pagetranschoices\space >>}
1030         \fi\fi\fi
1031     }
1032     \ifx\aeb@TranChoice\@empty\else
1033         \csarg{\xdef}{aebp@geTr@nsP@ge\arabic{page}}%
1034         {\noexpand\@nameuse{aeb@pagetransCodeL\aebp@PageTransCnt}}%
1035     \fi
1036 \endgroup
1037 }

```

`\setPageTransitionAt` Set the transition for a particular page

```

1038 \newcommand{\setPageTransitionAt}[2]{%
1039     \begingroup
1040     \aebp@Step@PageTransCnt
1041     \setkeys{aebTrans}{#2}%
1042     \edef\aeb@pagetranschoices{\aeb@Page@Dur
1043         /Trans <<\aeb@Trans@Dur\aeb@TranChoice>>}%
1044     \csarg{\xdef}{aeb@pagetransCodeL\aebp@PageTransCnt}{%
1045         \ifpdfmarkup
1046             \noexpand\literalps@out{\AEB@psMrk{ThisPage} <<
1047                 \aeb@pagetranschoices\space>> /PUT pdfmark}%
1048         \else\ifpdf

```

There may be open / close page actions already defined, so we combine previous page actions with current page actions. This seems necessary only for pdftex.

```

1049             \noexpand\global\noexpand\pdfpageattr=\noexpand\expandafter
1050             {\noexpand\the\noexpand\pdfpageattr\aeb@pagetranschoices}%
1051         \else\ifxetex
1052             \noexpand\immediate\noexpand\@pdfm@mark{put @thispage <<
1053                 \aeb@pagetranschoices\space>>}
1054         \fi\fi\fi
1055     }

```

```

1056 \ifx\ae@TranChoice\@empty\else
1057 \expandafter\ae@rse\expandafter{#1}%
1058 \edef\shakeandbake{%
1059 \noexpand\@for\noexpand\@mypage:=\the\ae@parsetoks}%
1060 \shakeandbake \do {%
1061 \csarg\xdef{ae@P@geTr@nsP@ge\@mypage}%
1062 {\noexpand\@nameuse{ae@pagetransCode\ae@PageTransCnt}}}%
1063 }%
1064 \fi
1065 \endgroup
1066 }

```

**Trans** The values of the Trans key. There are the 50 transitions effects as listed by `app.fs.transitions` plus one more called “NoTransition”. The pdfmark code accurately represents each of this options.

```

1067 \define@choicekey+{ae@Trans}{Trans}
1068 {NoTransition,UncoverLeft,UncoverRight,UncoverDown,UncoverUp,%
1069 UncoverLeftDown,UncoverLeftUp,UncoverRightDown,UncoverRightUp,%
1070 CoverLeft,CoverRight,CoverDown,CoverUp,CoverLeftDown,CoverLeftUp,%
1071 CoverRightDown,CoverRightUp,PushLeft,PushRight,PushDown,PushUp,%
1072 PushLeftDown,PushLeftUp,PushRightDown,PushRightUp,FlyInRight,%
1073 FlyInLeft,FlyInDown,FlyInUp,FlyOutRight,FlyOutLeft,FlyOutDown,%
1074 FlyOutUp,FlyIn,FlyOut,Blend,Fade,Random,Dissoolve,%
1075 GlitterRight,GlitterDown,GlitterRightDown,BoxIn,BoxOut,%
1076 BlindsHorizontal,BlindsVertical,SplitHorizontalIn,%
1077 SplitHorizontalOut,SplitVerticalIn,SplitVerticalOut,WipeLeft,%
1078 WipeRight,WipeDown,WipeUp,WipeLeftDown,WipeLeftUp,%
1079 WipeRightDown,WipeRightUp,Replace,ZoomInDown,ZoomInLeft,%
1080 ZoomInLeftDown,ZoomInLeftUp,ZoomInRight,ZoomInRightDown,%
1081 ZoomInRightUp,ZoomInUp,ZoomOutDown,ZoomOutLeft,ZoomOutLeftDown,%
1082 ZoomOutLeftUp,ZoomOutRight,ZoomOutRightDown,ZoomOutRightUp,%
1083 ZoomOutUp,CombHorizontal,CombVertical}[Replace]%
1084 {\edef\ae@TranChoice{\csname ae@Trans@#1\endcsname}%
1085 }\PackageWarning{ae}{Bad choice, try again}}
1086 \let\ae@TranChoice\@empty
1087 \define@key{ae@Trans}{TransDur}[]{\def\ae@arg{#1}\ifx\ae@arg\@empty
1088 \let\ae@Trans@Dur\@empty\else\def\ae@Trans@Dur{/D #1}\fi}
1089 \let\ae@Trans@Dur\@empty

```

**Speed** Length of duration of transition effect: Slow, Medium, Fast

```

1090 \define@choicekey+{ae@Trans}{Speed}[\val\nr]%
1091 {Slow,Medium,Fast}[Slow]{%dps/change 02/27/07
1092 \ifcase\nr\relax
1093 \let\ae@Trans@Speed\@empty\or
1094 \def\ae@Trans@Speed{/D 0.666656}\or
1095 \def\ae@Trans@Speed{/D 0.333328}\fi
1096 }\PackageWarning{ae}{Bad choice for Speed, permissible values
1097 are Slow, Medium and Fast. Try again}}
1098 \let\apb@apbTrans@Speed\@empty

```

**PageDur** The page duration key

```

1099 \define@key{aebTrans}{PageDur}[]{\def\ae@arg{#1}\ifx\ae@arg\@empty
1100 \let\ae@Page@Dur\@empty\else\def\ae@Page@Dur{/Dur #1}\fi}

```

The code for the values of the Trans key

```

1101 \let\ae@Page@Dur\@empty
1102 \def\ae@Trans@NoTransition{}
1103 \def\ae@Trans@SplitHorizontalIn{/Dm/H/M/I/S/Split}
1104 \def\ae@Trans@SplitHorizontalOut{/Dm/H/M/O/S/Split}
1105 \def\ae@Trans@SplitVerticalIn{/Dm/V/M/I/S/Split}
1106 \def\ae@Trans@SplitVerticalOut{/Dm/V/M/O/S/Split}
1107 \def\ae@Trans@BlindsHorizontal{/Dm/H/S/Blinds}
1108 \def\ae@Trans@BlindsVertical{/Dm/V/S/Blinds}
1109 \def\ae@Trans@BoxIn{/M/I/S/Box}
1110 \def\ae@Trans@BoxOut{/M/O/S/Box}
1111 \def\ae@Trans@WipeLeft{/Di 180/S/Wipe}
1112 \def\ae@Trans@WipeRight{/Di 0/S/Wipe}
1113 \def\ae@Trans@WipeDown{/Di 270/S/Wipe}
1114 \def\ae@Trans@WipeUp{/Di 90/S/Wipe}
1115 \def\ae@Trans@Dissolve{/S/Dissolve}
1116 \def\ae@Trans@GlitterRight{/Di 0/S/Glitter}
1117 \def\ae@Trans@GlitterDown{/Di 270/S/Glitter}
1118 \def\ae@Trans@GlitterRightDown{/Di 315/S/Glitter}
1119 \def\ae@Trans@Replace{/S/R}
1120 \def\ae@Trans@FlyInRight{/Di 0/M/I/S/Fly}
1121 \def\ae@Trans@FlyInLeft{/Di 180/M/I/S/Fly}
1122 \def\ae@Trans@FlyInDown{/Di 270/M/I/S/Fly}
1123 \def\ae@Trans@FlyInUp{/Di 90/M/I/S/Fly}
1124 \def\ae@Trans@FlyOutRight{/Di 0/M/O/S/Fly}
1125 \def\ae@Trans@FlyOutLeft{/Di 180/M/O/S/Fly}
1126 \def\ae@Trans@FlyOutDown{/Di 270/M/O/S/Fly}
1127 \def\ae@Trans@FlyOutUp{/Di 90/M/O/S/Fly}
1128 \def\ae@Trans@FlyIn{/M/I/S/Fly}
1129 \def\ae@Trans@FlyOut{/M/O/S/Fly}
1130 \def\ae@Trans@PushLeft{/Di 180/S/Push}
1131 \def\ae@Trans@PushRight{/Di 0/S/Push}
1132 \def\ae@Trans@PushDown{/Di 270/S/Push}
1133 \def\ae@Trans@PushUp{/Di 90/S/Push}
1134 \def\ae@Trans@CoverLeft{/Di 180/S/Cover}
1135 \def\ae@Trans@CoverRight{/Di 0/S/Cover}
1136 \def\ae@Trans@CoverDown{/Di 270/S/Cover}
1137 \def\ae@Trans@CoverUp{/Di 90/S/Cover}
1138 \def\ae@Trans@CoverLeftDown{/Di 225.0/S/Cover}
1139 \def\ae@Trans@CoverLeftUp{/Di 135.0/S/Cover}
1140 \def\ae@Trans@CoverRightDown{/Di 315.0/S/Cover}
1141 \def\ae@Trans@CoverRightUp{/Di 45.0/S/Cover}
1142 \def\ae@Trans@UncoverLeft{/Di 180/S/Uncover}
1143 \def\ae@Trans@UncoverRight{/Di 0/S/Uncover}
1144 \def\ae@Trans@UncoverDown{/Di 270/S/Uncover}
1145 \def\ae@Trans@UncoverUp{/Di 90/S/Uncover}
1146 \def\ae@Trans@UncoverLeftDown{/Di 225.0/S/Uncover}

```

```

1147 \def\aeB@Trans@UncoverLeftUp{/Di 135.0/S/Uncover}
1148 \def\aeB@Trans@UncoverRightDown{/Di 315.0/S/Uncover}
1149 \def\aeB@Trans@UncoverRightUp{/Di 45.0/S/Uncover}
1150 \def\aeB@Trans@Fade{/S/Fade}
1151 \def\aeB@Trans@Random{/S/Random}
1152 \def\aeB@Trans@Blend{/S/Blend}

New Transitions for Acrobat 8.0

1153 \def\aeB@Trans@PushLeftDown{%
1154     /S/Push/Directional/BiDir/Di 225.0/Curve/Easy}
1155 \def\aeB@Trans@PushLeftUp{%
1156     /S/Push/Directional/BiDir/Di 135.0/Curve/Easy}
1157 \def\aeB@Trans@PushRightDown{%
1158     /S/Push/Directional/BiDir/Di 315.0/Curve/Easy}
1159 \def\aeB@Trans@PushRightUp{%
1160     /S/Push/Directional/BiDir/Di 45.0/Curve/Easy}
1161 \def\aeB@Trans@WipeLeftDown{%
1162     /S/Wipe/Directional/BiDir/Di 225/Curve/Easy}
1163 \def\aeB@Trans@WipeLeftUp{%
1164     /S/Wipe/Directional/BiDir/Di 135/Curve/Easy}
1165 \def\aeB@Trans@WipeRightDown%
1166     {/S/Wipe/Directional/BiDir/Di 315/Curve/Easy}
1167 \def\aeB@Trans@WipeRightUp{%
1168     /S/Wipe/Directional/BiDir/Di 45/Curve/Easy}
1169 \def\aeB@Trans@ZoomInDown{%
1170     /S/ZoomIn/Directional/BiDir/Di 270.0/Curve/Easy}
1171 \def\aeB@Trans@ZoomInLeft{%
1172     /S/ZoomIn/Directional/BiDir/Di 180.0/Curve/Easy}
1173 \def\aeB@Trans@ZoomInLeftDown{%
1174     /S/ZoomIn/Directional/BiDir/Di 225.0/Curve/Easy}
1175 \def\aeB@Trans@ZoomInLeftUp{%
1176     /S/ZoomIn/Directional/BiDir/Di 135.0/Curve/Easy}
1177 \def\aeB@Trans@ZoomInRight{%
1178     /S/ZoomIn/Directional/BiDir/Di 0.0/Curve/Easy}
1179 \def\aeB@Trans@ZoomInRightDown{%
1180     /S/ZoomIn/Directional/BiDir/Di 315.0/Curve/Easy}
1181 \def\aeB@Trans@ZoomInRightUp{%
1182     /S/ZoomIn/Directional/BiDir/Di 45.0/Curve/Easy}
1183 \def\aeB@Trans@ZoomInUp{%
1184     /S/ZoomIn/Directional/BiDir/Di 90.0/Curve/Easy}
1185 \def\aeB@Trans@ZoomOutDown{%
1186     /S/ZoomOut/Directional/BiDir/Di 270.0/Curve/Easy}
1187 \def\aeB@Trans@ZoomOutLeft{%
1188     /S/ZoomOut/Directional/BiDir/Di 180.0/Curve/Easy}
1189 \def\aeB@Trans@ZoomOutLeftDown{%
1190     /S/ZoomOut/Directional/BiDir/Di 225.0/Curve/Easy}
1191 \def\aeB@Trans@ZoomOutLeftUp{%
1192     /S/ZoomOut/Directional/BiDir/Di 135.0/Curve/Easy}
1193 \def\aeB@Trans@ZoomOutRight{%
1194     /S/ZoomOut/Directional/BiDir/Di 0.0/Curve/Easy}

```

```

1195 \def\aeB@Trans@ZoomOutRightDown{%
1196     /S/ZoomOut/Directional/BiDir/Di 315.0/Curve/Easy}
1197 \def\aeB@Trans@ZoomOutRightUp{%
1198     /S/ZoomOut/Directional/BiDir/Di 45.0/Curve/Easy}
1199 \def\aeB@Trans@ZoomOutUp{%
1200     /S/ZoomOut/Directional/BiDir/Di 90.0/Curve/Easy}
1201 \def\aeB@Trans@CombHorizontal{%
1202     /S/Comb/Directional/BiDir/Dm/H/Curve/Easy}
1203 \def\aeB@Trans@CombVertical{%
1204     /S/Comb/Directional/BiDir/Dm/V/Curve/Easy}

1205 % End of aeBpro segment
1206 \</aeBpro>
1207 \<*/fsdefs>

```

### 8.3 JS for Fullscreen Support

The “Presentation Defaults” are used in conjunction with the `\setDefaultFS` for setting the defaults of the application. These settings can be viewed under the `Edit > Preferences > General > Full Screen`. This script save the viewers preferences, sets them according the needs of the document author, and upon closing the document, resets the original preferences. The last task is performed by the `willClose` environment.

```

1208 \begin{insDLJS}[_fsDefaults]{fsdef}{AeB Pro: Presentation Defaults}
1209 var _fsDefaults = true;
1210 %if ( typeof global.fsexec == "undefined" )
1211 if ( typeof fsexec == "undefined" )
1212 {
1213     try {
1214         var fsexec = true;
1215         %
1216         global.fsexec = true;
1217         var aeBdefaultTransition = app.fs.defaultTransition;
1218         var aeBbackgroundColor = app.fs.backgroundColor;
1219         var aeBloop = app.fs.loop;
1220         var aeBtimeDelay = app.fs.timeDelay;
1221         var aeBuseTimer = app.fs.useTimer
1222         var aeBusePageTiming = app.fs.usePageTiming;
1223         var aeBclickAdvances = app.fs.clickAdvances;
1224         var aeBcursor = app.fs.cursor;
1225         var aeBescapeExits = app.fs.escapeExits;
1226 \aeB@fsTran%
1227 \aeB@fsBGColor%
1228 \aeB@fsLoop%
1229 \aeB@fsclickAdv%
1230 \aeB@fscursor%
1231 \aeB@fstimeDelay%
1232 \aeB@fsuseTimer%
1233 \aeB@fsusePageTiming%
1234 \aeB@fsEscape%

```

```

1234 \aeb@fsFS%
1235   } catch(e) {}
1236 }
1237 \end{insDLJS}

```

This is a series of script lines that will be inserted into the `willClose` section of the document. These will attempt to restore the user's full screen defaults.

```

1238 \begin{fs@willClose}
1239 try {
1240   delete global.fsexec;
1241   app.fs.defaultTransition = aebdefaultTransition;
1242   app.fs.backgroundColor = aebbackgroundColor;
1243   app.fs.loop = aebloop;
1244   app.fs.timeDelay = aebtimeDelay;
1245   app.fs.useTimer = aebuseTimer;
1246   app.fs.usePageTiming = aebusePageTiming;
1247   app.fs.clickAdvances = aebclickAdvances;
1248   app.fs.cursor = aebcursor;
1249   app.fs.escapeExits = aebescapeExits;
1250 } catch(e) { console.println("Could not reset one of the defaults"); }
1251 \end{fs@willClose}

1252 % End of fsdefjs segment
1253 \</fsdefjs>
1254 \<*aebpro>

```

## 9 Set Page Actions

`addJSToPageOpen` The document author can add to the default page open action through this environment. Use this environment between slides, effective the next slide. The `addJSToPageOpenAt` first slide (possibly generated by `\maketitle`) is a special case: When setting the `addJSToPageClose` open and close page actions for the first page, these environments *must go* in the `addJSToPageCloseAt` preamble.

```

1255 \def\canceleveryPageOpen{\global\@everyPageOpenfalse
1256   \global\let\every@PageOpen\@empty}
1257 \def\canceleveryPageClose{\global\@everyPageClosefalse
1258   \global\let\every@PageClose\@empty}
1259 \newenvironment{everyPageOpen}[1] []
1260 {\global\@everyPageOpentruel\@defineJS[#1]{\every@PageOpen}}
1261 {\end@defineJS
1262   \ifnum\value{page}=\@ne
1263     \gdef\ap@AddtoOpenJS{\thisPageAction{\JS{\every@PageOpen}}{}}}%
1264   \expandafter\aftergroup\expandafter\ap@AddtoOpenJS\fi
1265 }
1266 \newenvironment{everyPageClose}[1] []
1267 {\global\@everyPageClosetrue\@defineJS[#1]{\every@PageClose}}
1268 {\end@defineJS
1269   \ifnum\value{page}=\@ne
1270     \gdef\ap@AddtoCloseJS{\thisPageAction{}{\JS{\every@PageClose}}}%

```

```

1271 \expandafter\aftergroup\expandafter\ap@AddtoCloseJS\fi
1272 }
1273 \let\every@PageOpen\@empty
1274 \let\every@PageClose\@empty
1275 \newenvironment{addJSToPageOpen}[1] []
1276 {\@defineJS[#1]{\addJSToPageOpen}}{\end@defineJS
1277 \ifnum\value{page}=\@ne
1278 \gdef\ap@AddtoOpenJS{\thisPageAction{\JS{\addJSToPageOpen}}{}}%
1279 \expandafter\aftergroup\expandafter\ap@AddtoOpenJS\fi
1280 }
1281 \newenvironment{addJSToPageOpenAt}[2] []
1282 {\@defineJS[#1]{addJSToPageOpenAt}{#2}}{\end@defineJS

```

The addJSToPageOpenAt and addJSToPageCloseAt can now be used in the preamble on the first page.

```

1283 \ifnum\value{page}=\@ne\iffrstPageSpec\aebp@YES
1284 \thisPageAction{\JS{\@nameuse{addJSToPageOpenAt1}}}{\fi\fi
1285 }
1286 \newenvironment{addJSToPageClose}[1] []
1287 {\pdfSpacesOff\@defineJS[#1]{\addJSToPageClose}}{\end@defineJS
1288 \ifnum\value{page}=\@ne
1289 \gdef\ap@AddtoCloseJS{%
1290 \thisPageAction{\JS{\addJSToPageClose}}}%
1291 \expandafter\aftergroup\expandafter\ap@AddtoCloseJS\fi
1292 }
1293 \newenvironment{addJSToPageCloseAt}[2] []
1294 {\pdfSpacesOff\@defineJS[#1]{addJSToPageCloseAt}{#2}}{\end@defineJS
1295 \ifnum\value{page}=\@ne\iffrstPageSpec\aebp@YES
1296 \thisPageAction{\JS{\@nameuse{addJSToPageCloseAt1}}}{\fi\fi
1297 }
1298 \let\addJSToPageOpen\@empty
1299 \let\addJSToPageClose\@empty

```

\aebp@PageOpenCloseHandler This command gets expanded by \AddToShipoutPicture for each page. Here we define how our page actions are to be inserted.

```

1300 \def\aebp@PageOpenCloseHandler{%
1301 \@ifundefined{addJSToPageOpenAt\arabic{page}}%
1302 {\global\let\thisOpenPageActionAt\@empty}%
1303 {\gdef\thisOpenPageActionAt{%
1304 \@nameuse{addJSToPageOpenAt\arabic{page}}}%
1305 \@ifundefined{addJSToPageCloseAt\arabic{page}}%
1306 {\global\let\thisClosePageActionAt\@empty}%
1307 {\gdef\thisClosePageActionAt{%
1308 \@nameuse{addJSToPageCloseAt\arabic{page}}}%
1309 \ifx\addJSToPageOpen\@empty\def\x{}\else\def\x{^J}\fi
1310 \ifx\addJSToPageClose\@empty\def\y{}\else\def\y{^J}\fi
1311 \aebPageAction{\thisOpenPageActionAt\x\addJSToPageOpen}%
1312 {\thisClosePageActionAt\y\addJSToPageClose}%
1313 \global\let\addJSToPageOpen\@empty
1314 \global\let\addJSToPageClose\@empty

```



```

1315 }
      (2016/05/15) Added \aebp@PageTransHandler to handle the page transitions.
1316 \def\aebp@PageTransHandler{\ifundefined{aebp@geTr@nsP@ge\arabic{page}}
1317   {\@nameuse{aebp@geTr@nsP@ge\arabic{page}}}}
      The two handlers are executed on each page
1318 \AddToShipoutPicture{\aebp@PageOpenCloseHandler\aebp@PageTransHandler}

```

**@@defineJS** is a variation of `\@defineJS`, as defined in the `insdljs` package. This version, at the end of the environment, calls the `\aebp@rse` command to parse the complex command argument, then makes a series of definitions to be used in the open/close page events.

```

1319 \newenvironment{@@defineJS}[3][
1320   {%
1321     \@initdfnJS
1322     \gdef\argType{#2}\gdef\defineJSArg{#3}\JStoks={}\dlfortypesetfalse
1323     \@vrbsetupdfnJS\b@vrbdfnJS{#1}%
1324   }{%
1325     \expandafter\aebp@rse\expandafter{\defineJSArg}%
1326     \edef\sh@keandbake{\noexpand
1327       \@for\noexpand\@mytoks:=\the\aebp@parsetoks}%
1328     \aeb@parsecnt=\z@\sh@keandbake\do{%
1329       \ifnum\aeb@parsecnt=\z@
1330         \edef\firsttok{\@mytoks}%
1331         \csarg\xdef{\argType\@mytoks}{\the\JStoks}%
1332       \else
1333         \edef\letsmakelet{\global\let\expandafter\noexpand
1334           \csname\argType\@mytoks\endcsname=\expandafter\noexpand
1335           \csname\argType\firsttok\endcsname}\letsmakelet
1336       \fi
1337       \advance\aeb@parsecnt\@ne
1338     }%
1339   }

1340 \newif\ifaeb@PageOpen
1341 \newif\ifaeb@PageClose
1342 \newif\if@everyPageOpen\@everyPageOpenfalse
1343 \newif\if@everyPageClose\@everyPageClosefalse

```

**\aebPageAction** This is used internally, but can be used by the document author. The command creates the code to set an open page and a close page action. The first (optional) argument is a the page number (base 1), this optional argument defaults to the current page if no page number is given. The second and third arguments are the open and closed JavaScript actions.

If the document author uses this command, he may overwrite actions created by the `addJSTo...` environments. See `\aeb@addToPageOpenCloseActions` for usage by this package.

```

1344 \newcommand\aebPageAction[3][ThisPage]{%

```

(2021/02/07) The core L<sup>A</sup>T<sub>E</sub>X `\@activechar@info` macro was breaking the page open and close events by inserting an unwanted message into the code. We redefine this macro to remove the message and insert `\pdfSP`, which either a normal space or `\pdfSPDef` depending on whether `\pdfSpacesOff` or `\pdfSpacesOn`, respectively, is operational. This change is local (I hope).

```

1345 \def\@activechar@info{\pdfSP\@gobbletwo}\let\ispageactions\@ebp@Zero
1346 \@eb@PageOpenfalse\@eb@PageClosefalse
1347 \def\@aebOpen{#2}%
1348 \if1\@aebOpen1%
1349   \if@everyPageOpen
1350     \@eb@PageOpentruel\let\ispageactions\@ebp@One
1351     \def\@aebOpen{/O << \JS{\every@PageOpen#2} >>}\fi
1352 \else
1353   \let\ispageactions\@ebp@One
1354   \@eb@PageOpentruel
1355   \def\@aebOpen{/O << \JS{\every@PageOpen^^J#2} >>}%
1356 \fi
1357 \def\@aebClose{#3}%
1358 \if1\@aebClose1%
1359   \if@everyPageClose
1360     \@eb@PageClosetruel\let\ispageactions\@ebp@One
1361     \def\@aebClose{/C << \JS{\every@PageClose#3} >>}\fi
1362 \else
1363   \let\ispageactions\@ebp@One
1364   \@eb@PageClosetruel
1365   \def\@aebClose{/C << \JS{\every@PageClose^^J#3} >>}%
1366 \fi
1367 \if@everyPageOpen\@eb@PageOpentruel\let\ispageactions\@ebp@One\fi
1368 \if@everyPageClose\@everyPageClosetruel\let\ispageactions\@ebp@One\fi
1369 \if\ispageactions\@ebp@One
1370   \xdef\@ebp@PageActionCode{/AA <<
1371     \ifaeb@PageOpen\@aebOpen\space\fi
1372     \ifaeb@PageClose\@aebClose\space\fi >>}%

```

For page 1, the open page is handled by `insdljs`

```

1373 \ifnum\value{page}=\@ne\else
1374 \ifpdfmarkup\literalps@out{\AEB@psMrk{#1} <<
1375   \@ebp@PageActionCode\space >> /PUT pdfmark}\else
1376 \ifpdf\global\pdfpageattr=\expandafter
1377   {\@ebp@PageActionCode}\else
1378 \ifxetex\@pdfm@mark{put @thispage <<
1379   \@ebp@PageActionCode\space>>}\fi\fi\fi
1380 \fi
1381 \fi
1382 }

```

`\atPage` `\atPage` is a modification of `\pageref`, so its value will always be an integer. returned in the token register `\@eb@parsetoks`. Used in the arguments of such environments as `addJSToPageOpenAt` and `addJSToPageCloseAt`.

```

1383 \def\aeboxiii{\expandafter\expandafter\expandafter}
1384 \let\ahrefexafter\aeboxiii
1385 \def\atPage#1{\@ifundefined{r#1}{1}
1386   {\aeboxiii\@secondoffive\csname r#1\endcsname}}
1387 \def\labelRef#1{\@ifundefined{r#1}{Doc-Start}
1388   {\aeboxiii\@fourthoffive\csname r#1\endcsname}}

```

The command `\aebp@rse` parses such arguments as

```
{1,3,4-6,8-12,15,\atPage{firstlabel}-\atPage{secondlabel},30-}
```

The expanded list is returned in the token register `\aeb@parsetoks`.

```

1389 \newtoks\aeb@parsetoks
1390 \newcount\aeb@parsecnt
1391 \def\ap@gobtonil#1\@nil{
1392 \def\ap@gettonil#1\@nil{\def\to@nilarg{#1}}
1393 \def\ap@ifspc{\ifx\@let@token\@sptoken
1394   \let\ap@next\ap@xifspc\else
1395   \let\ap@next\ap@gettonil\fi\ap@next}
1396 \begingroup
1397 \def\:\{\ap@xifspc}
1398 \expandafter\gdef\:\{\futurelet\@let@token\ap@ifspc}
1399 \endgroup

```

`\ap@strpspcs` strips any spaces, usage: `\expandafter\ap@strpspcs\cmd\@nil` where `\cmd` is a command that expands to text; the text is returned in `\to@nilarg` without any leading spaces.

```

1400 \def\ap@strpspcs{\futurelet\@let@token\ap@ifspc}
1401 \def\ap@pgrngLabelsInPream{\@ifundefined{aebLastPage}
1402   {\PackageWarning{aeb_pro}{The \string\atPage\space
1403     command is not supported\MessageBreak in the preamble}}{}}
1404 \def\aeboxremovelastcomma#1,\@nil{\aeb@parsetoks={#1}}
1405 \def\aeboxp@rse#1{\global\let\frstPageSpec\aebox@NO
1406   \aeb@parsetoks={}\aeb@parse#1,\relax}
1407 \def\aebox@parse{\@ifnextchar\relax
1408   {\expandafter\aeboxremovelastcomma\the\aeb@parsetoks\@nil}
1409   {\aeb@@@parse}}
1410 \def\aebox@@@parse#1,{\aeb@@@parse#1-\@nil}
1411 \def\aebox@@@parse#1-#2\@nil{%
1412   \def\argi{#2}\ifx\argi\@empty\def\aebox@next{\aeb@@@parse{#1}}\else
1413     \def\aebox@next{\aeb@@@parse#1-#2}\fi\aebox@next}
1414 \def\aebox@@@parse#1{%single page #1
1415   \@ifnextchar\atPage{\ap@pgrngLabelsInPream\ap@gobtonil}
1416     {\ap@gobtonil}#1\@nil
1417   \aeb@parsecnt=#1\relax
1418   \edef\edefaeb@parsetoks{\the\aeb@parsetoks\the\aeb@parsecnt,}%
1419   \aeb@parsetoks=\expandafter{\edefaeb@parsetoks}%

```

The `\frstPageSpec` signals that page 1 is specified in the page specification argument.

```
1420   \ifnum\aeb@parsecnt=1\relax\global\let\frstPageSpec\aebox@YES\fi
```

```

1421 \aeb@parse}
1422 \def\aeb@@@parse#1-#2-{\def\aeb@lowerlimit{#1}\def\aeb@upperlimit{#2}%
1423 % range argument from #1 to \csname aebLastPage\endcsname
1424 \ifx\aeb@upperlimit\@empty
1425 \ifundefined{aebLastPage}{\aeb@parsecnt=#1}
1426 {\expandafter\aeb@parsecnt\expandafter=
1427 \csname aebLastPage\endcsname}%
1428 \else % page range #1 to #2, switch the two if #2 < #1
1429 \aeb@parsecnt=\aeb@lowerlimit\relax
1430 \edef\aeb@lowerlimit{\the\aeb@parsecnt}
1431 \aeb@parsecnt=\aeb@upperlimit\relax
1432 \ifnum\aeb@parsecnt<\aeb@lowerlimit\relax
1433 \aeb@parsecnt=\aeb@lowerlimit\relax
1434 \def\aeb@lowerlimit{#2}%
1435 \def\aeb@upperlimit{#1}%
1436 \fi
1437 \fi
1438 \@ifnextchar\atPage{\ap@pgrngLabelsInPream\ap@gobtonil}
1439 {\ap@gobtonil}#1\@nil
1440 \@ifnextchar\atPage{\ap@pgrngLabelsInPream\ap@gobtonil}
1441 {\ap@gobtonil}#2\@nil
1442 \advance\aeb@parsecnt1\relax
1443 \edef\aeb@upperlimit{\the\aeb@parsecnt}%
1444 \aeb@parsecnt=\aeb@lowerlimit\relax
    If page 1 is there, is will be as the lower limit. The \frstPageSpec signals that
    page 1 is specified in the page specification argument.
1445 \ifnum\aeb@parsecnt=1\relax\global\let\frstPageSpec\aebp@YES\fi
1446 \loop
1447 \edef\edefaeb@parsetoks{\the\aeb@parsetoks\the\aeb@parsecnt,}%
1448 \aeb@parsetoks=\expandafter{\edefaeb@parsetoks}%
1449 \advance\aeb@parsecnt1\relax
1450 \ifnum\aeb@parsecnt < \aeb@upperlimit\relax\repeat
1451 \aeb@parse}

```

## 10 Set Document Actions

**willClose** We add a `willClose` environment, not really meant for the document author to use. This environment is used with `\setDefaultFS` for saving setting and restoring the application preferences for full screen.

**willPrint** `\@ifpackageloaded{insdljs}{%`

**didPrint** `\newenvironment{fs@willClose}`

```

1452 \newenvironment{fs@willClose}
1453 {\@defineJS{fs@will@Close}}
1454 {\end@defineJS}
1455 \let\fs@will@Close\empty
1456 }{%
1457 \newenvironment{willClose}
1458 {\@defineJS{will@Close}}
1459 {\end@defineJS}
1460 }

```

```

1461 \let\will@Close@empty
1462 \newenvironment{fs@will@Close}
1463 {\@defineJS{fs@will@Close}}
1464 {\end@defineJS}
1465 \let\fs@will@Close@empty
1466 \newenvironment{will@Save}
1467 {\@defineJS{will@Save}}
1468 {\end@defineJS}
1469 \let\will@Save@empty
1470 \newenvironment{did@Save}
1471 {\@defineJS{did@Save}}
1472 {\end@defineJS}
1473 \let\did@Save@empty
1474 \newenvironment{will@Print}
1475 {\@defineJS{will@Print}}
1476 {\end@defineJS}
1477 \let\will@Print@empty
1478 \newenvironment{did@Print}
1479 {\@defineJS{did@Print}}
1480 {\end@defineJS}
1481 \let\did@Print@empty
1482 }

```

In addition to the provided environments, there are developer hooks, the package developer can use to insert page actions, in addition to what the package user may do. Now put the `willClose` JS into the document `Catalog`. The developer's actions will possibly overwrite that of the user; under the theory that the developer's actions are critical to the functioning of the package. The hooks are

```

1483 \let\developer@will@Close@empty
1484 \let\developer@will@Save@empty
1485 \let\developer@did@Save@empty
1486 \let\developer@will@Print@empty
1487 \let\developer@did@Print@empty

```

(2016/05/15) modify to include non-Distiller workflows.

```

1488 \newcommand\@setDocActions{%
1489   \edef\test@will@Close{%
1490     \fs@will@Close\will@Close\developer@will@Close}%
1491   \edef\test@will@Save{\will@Save\developer@will@Save}%
1492   \edef\test@did@Save{\did@Save\developer@did@Save}%
1493   \edef\test@will@Print{\will@Print\developer@will@Print}%
1494   \edef\test@did@Print{\did@Print\developer@did@Print}%
1495   \def\@ebp@DocActionsCode{AA << %
1496     \ifx\test@will@Close@empty\else/WC <<
1497       \JS{fs@will@Close\will@Close\developer@will@Close} >>\fi
1498     \ifx\test@will@Save@empty\else/WS <<
1499       \JS{will@Save\developer@will@Save} >>\fi
1500     \ifx\test@did@Save@empty\else/DS <<
1501       \JS{did@Save\developer@did@Save} >>\fi
1502     \ifx\test@will@Print@empty\else/WP <<

```

```

1503         \JS{\will@Print\developer@will@Print} >>\fi
1504     \ifx\test@did@Print\@empty\else/DP <<
1505         \JS{\did@Print\developer@did@Print} >>\fi
1506     >>
1507 }
1508 \ifpdfmarkup\literalps@out{\AEB@psMrk{Catalog}
1509     << \aebp@DocActionsCode\space>> /PUT pdfmark}\else
1510 \ifpdf\pdfcatalog{\aebp@DocActionsCode}\else
1511 \ifxetex\immediate
1512     \@pdfm@mark{put @catalog <<\aebp@DocActionsCode\space>>}\fi
1513 \fi\fi
1514 }
1515 \AtBeginDocument{\@setDocActions}

```

## 11 In support of attachments

### 11.1 Processing the attachsource and attachments key

If the value of either the `attachsource` or `attachments` keys are non empty, the values are contained in the macros `\aeb@attachsource` and `\aeb@attachments`. If either of these text macros is different from `\empty`, we need to write out a cut file that lists the requested attachments, then input the cut file back in at the end of the package.

`\aref` This command, whose argument is *delimited by parentheses*, is to be used with `\inputDataObject`, defined below, to expand the value of the `cName` key, when the label is defined by `\labelName`. It is also used by this package to set the description of the attachment.

```

1516 \def\aref(#1){\csname aeb@aDescript@#1\endcsname}

```

The next command `\aeb@initAttachWrite` we prepare to write by opening a stream, and starting the `execJS` environment, but only if we have an attachment request.

```

1517 \def\aeb@initAttachWrite{%
1518     \newwrite \aeb@attachmentswrite
1519     \immediate\openout \aeb@attachmentswrite \jobname_attach.cut
1520     \let\verbatim@out\aeb@attachmentswrite
1521     \aeb@IWVO{\string\begin{execJS}{aebpro_attach}}%
1522     {\lccode'P='\{\lowercase{\aeb@IWVO{try P}}}%
1523 }

```

**Optional attachments.** Support for optional attachments.

`\prjInputUser` User interface to project for `\input` and `\include`. These commands may be redefined, but they need to include the appropriate input/include command in them. The `\addtoOptAttachments` command is a way of optionally attaching a file without the input.

```

1524 \newcommand{\prjInputUser}[1]{\input{#1}}
1525 \newcommand{\prjIncludeUser}[1]{\include{#1}}
1526 \newcommand{\addtoOptAttachments}[1]{%
1527     \g@addto@macro\AP@addtoAttachments{, #1}}
1528 \newcommand{\getNameFromFileName}[1]{\@nameuse{descrip4#1}}

```

The main code for optional attachments.

```
1529 \def\AP@addtoAttachments{}
```

The attachments are listing in a comma-delimited list. If there is a space following a comma, it may appear as part of the path of the next attachment. So, we'll get JavaScript to strip away any leading spaces to avoid the need to instruct the user to insert a % if the attachment list is on multiple lines.

```

1530 \def\jsstrsps{replace(/^\string\s*/, "")}
1531 \def\jsstrdotsp{replace(/\/string\.\string\s*/g, ".")}
1532 \ifoptattachments
1533     \def\AP@writeOptAttach{%
1534         \newwrite\AP@OptAttachlist
1535         \immediate\openout \AP@OptAttachlist \jobname_oa.cut

```

Define the optional attachments list command, \APOptAttachments.

```

1536         \immediate\write\AP@OptAttachlist{\string\def\string
1537             \APOptAttachments{\AP@addtoAttachments}}
1538         \immediate\closeout\AP@OptAttachlist}

```

Write the optional attachments list (\APOptAttachments) at the end of the document.

```

1539     \AtEndDocument{\AP@writeOptAttach}
1540 %     \def\AP@addtoAttachments{}
1541 \def\AP@pathInput#1{%
1542     \addtoOptAttachments{#1}\prjInputUser{#1}}
1543 \def\AP@pathInclude#1{%
1544     \addtoOptAttachments{#1.tex}\prjIncludeUser{#1}}

```

Input the optional attachments list (\APOptAttachments) here.

```

1545     \InputIfFileExists{\jobname_oa.cut}{%
1546         \def\AP@next{\edef\aeb@attachments
1547             {\aeb@attachments\APOptAttachments}}}%
1548     }{\let\AP@next\relax}
1549     \AP@next
1550 \else

```

When !optattachments is taken, we use the user commands \prjInputUser and \prjIncludeUser versions.

```

1551     \def\AP@pathInput#1{\prjInputUser{#1}}
1552     \def\AP@pathInclude#1{\prjIncludeUser{#1}}
1553 \fi

```

\prjinput These are the main commands for inputting project files where the files themselves  
\prjinclude are optionally attached.

```

1554 \ifoptattachmentsTaken
1555     \def\prjinput{\hyper@normalise\AP@pathInput}

```

```
1556 \def\prjinclude{\hyper@normalise\AP@pathInclude}
```

The `\include` command makes the following comparison.

```
\@for\reserved@a:=\@partlist\do
  {\ifx\reserved@a\reserved@b\@tempwatrue\fi}%
```

The elements from `\@parlist` they come from the `\includeonly` command. The argument of `\include` has been normalized using `\hyper@normalise` but the file names listed in `\includeonly` have not. So the comparison can fail. We normalize the file list as well.

When the `optattachments` is present in the option list, the definition of `\optattachmentsTaken` is `\relax`, hence, the redefinition occurs.

```
1557 \let\AP@includeonly\includeonly
1558 \def\includeonly{\hyper@normalise\AP@includeonly}
1559 \else
1560 \def\prjinput{\prjInputUser}
1561 \def\prjinclude{\prjIncludeUser}
1562 \fi
```

**Attaching files.** If either the two attachments keys is used, then we execute `\aeb@initAttachWrite`.

```
1563 \ifx\aeb@attachsource\@empty
1564 \ifx\aeb@attachments\@empty
1565 \else\aeb@initAttachWrite\fi
1566 \else
1567 \ifundefined{aeb@attachmentswrite}
1568 {\aeb@initAttachWrite}{\}
1569 \fi
```

If `\aeb@attachsource` is non empty, go through the comma delimited list, and write to the cut file we just opened.

Some commands for setting `cName` and description.

```
1570 \def\ap@cName(#1){\csname aeb@aName@attach#1\endcsname}
1571 \def\ap@cNamexdef(#1){\expandafter
1572 \xdef\csname aeb@aName@attach#1\endcsname}
1573 \def\ap@cDescript(#1){\csname aeb@aDescript@attach#1\endcsname}
1574 \def\ap@cDescriptSource(#1){\csname aeb@aDescript@#1\endcsname}
1575 \def\ap@cDescriptxdef(#1){\expandafter
1576 \xdef\csname aeb@aDescript@attach#1\endcsname}
1577 \def\ap@cDescriptSourcexdef(#1){\expandafter
1578 \xdef\csname aeb@aDescript@#1\endcsname}
```

We now process the list from the `attachsource` key.

```
1579 \ifx\aeb@attachsource\@empty\else
1580 \edef\temp@expand{\noexpand\@for\noexpand\@tmp:=\aeb@attachsource}
1581 \temp@expand\do{%
1582 \ifx\@tmp\@empty\else
1583 \expandafter\ap@strpspc\@tmp\@nil\edef\@tmp{\to@nilarg}%
1584 \ap@cDescriptSourcexdef(\@tmp){\jobname.\@tmp}%
```



```

1585         \aeb@IWV0{\space\space\space\space
1586         var retn=\string\importDataObject({%
(2015/07/01) Changed cName from \jobname.\@tmp to just \@tmp.
1587         cName: "\@tmp",
1588         cDIPath: "\jobname.\@tmp".\jsstrdotsp});}%
1589         \aeb@IWV0{\space\space\space\space
1590         if ( (app.viewerVersion>7) && retn )
(2015/07/01) Changed "\jobname.\@tmp" to just "\@tmp" within argument of
this.getDataObject.
1591         this.getDataObject("\@tmp").description=%
1592         "\string\ap@cDescriptSource(\@tmp)".\jsstrdotsp};}%
1593         \expandafter
1594         \edef\cename aeb@aName@\@tmp\endcsname{\jobname.\@tmp}%
1595     \fi
1596 }%
1597 \fi

```

Similarly, if \aeb@attachments is non empty, go through the comma delimited list, and write to the cut file we just opened.

```

1598 \ifx\aeb@attachments\@empty\else
1599 \typeout{\string\aeb@attachments: \aeb@attachments}%
1600 \aeb@parsecnt=0
1601 \edef\temp@expand{\noexpand\@for\noexpand\@tmp:=\aeb@attachments}
1602 \temp@expand\do
1603 {%
1604     \ifx\@tmp\@empty\else
1605         \expandafter\ap@strpspcs\@tmp\@nil\edef\@tmp{\to@nilarg}%
1606         \advance\aeb@parsecnt1\relax

```

Set default definitions of the name and description

```

1607         \ap@cName\edef(\the\aeb@parsecnt){attach\the\aeb@parsecnt}%
1608         \ap@cDescriptxdef(\the\aeb@parsecnt)%
1609         {AeB Attachment \the\aeb@parsecnt}%
1610         \aeb@IWV0{\space\space\space\space
1611         var retn=\string\importDataObject({%

```

(2015/07/01) Remove \aref in cName

```

1612         cName: "\string\ap@cName(\the\aeb@parsecnt)",%
1613         cDIPath: "\@tmp".\jsstrsps});}%
1614         \aeb@IWV0{\space\space\space\space
1615         if ( (app.viewerVersion>7) && retn )

```

(2015/07/01) Remove \aref in first argument

```

1616         this.getDataObject(%
1617         "\string\ap@cName(\the\aeb@parsecnt)").description=%
1618         "\string\ap@cDescript(\the\aeb@parsecnt)";}%
1619         \expandafter

```

We associate the filename (path) with the label cName. This makes it convenient to reference an attachment. We can say \@usename{descrip4<name>}. Also

defined a convenience command `\getNameFromFileName` for that purpose; for example, `\getNameFromFileName{<name>}`.

```
1620         \expandafter\xdef\csname descrip4\@@tmp\endcsname
1621             {\noexpand\ap@cname(\the\ae@parsecnt)}%
1622     \fi
1623 }
1624 \fi
```

Now, if `\ae@attachmentswrite` is defined, which means we have attachments and we opened a stream, we'll close off the environment, and close the file.

```
1625 \@ifundefined{ae@attachmentswrite}{%
1626 {%
1627     {\lccode'P='}\def\jsMsg{"Attachments error: "
1628         + e.toString()}\lowercase{\ae@IWV0{P catch(e)
1629             { console.println( \jsMsg\space); }}}}%
1630     \ae@IWV0{\string\end{execJS}}
1631     \immediate\closeout \ae@attachmentswrite
1632 }
```

Create a little macro to input the file `\jobname.attach.cut` at the end of the package.

```
1633 \def\ae@inputAttachments{\@ifundefined{ae@attachmentswrite}{%
1634     {\InputIfFileExists{\jobname.attach.cut}}%
1635     {\typeout{Inputting attachments file}}%
1636     {\typeout{Cannot find attachments file}}}%
1637 }
```

## 11.2 Creating a PDF Package

Beginning with Version 8, there is a special "Package" interface to access attachments. The commands in this section support creating a PDF Package using the specified attachments.

**viewmode** The `viewmode` has values of `details`, `tile`, `layout`, and `hidden`.

```
1638 \define@choicekey+{ae@pdfpack}{viewmode}[\val\nr]%
1639     {details,tile,layout,hidden}{%
1640     \ifcase\nr\relax
1641         \def\ae@pdfpack@viewmode{/D}\or
1642         \def\ae@pdfpack@viewmode{/T}\or
1643         \def\ae@pdfpack@viewmode{/T}\or
1644         \def\ae@pdfpack@viewmode{/H}\fi
1645 }{}
```

**initview** The `initview` of the package. The value of `initview` is a label name of one of the attached files.

```
1646 \define@key{ae@pdfpack}{initview}[]{\def\ae@pdfpack@initview{#1}}
1647 \let\ae@pdfpack@initview\empty
```

**\earlyAttachForPkgs** Use this command to package the attachments to the parent document.

```
1648 \newcommand{\earlyAttachForPkgs}{%
```

(2015/07/11) Recent versions of Acrobat DC seem not to support the Hidden view. There is no difference between Hidden and Tiles, so we use 'CollectionPreview' menu item to force the viewer into Preview mode, which I'm assuming is the same as Hidden mode.

```

1649 if (this.collection!=null&&this.collection.initialView == "Hidden")
1650 var hideto=app.setTimeout("app.clearTimeOut(hideto);"
1651   +"app.execMenuItem('CollectionPreview');" ,1);
1652 \if@gopro
1653 if(typeof _aebpro_attach == "undefined")
1654 ( app.viewerVersion > 8 ) ? %
1655 aebTrustedFunctions(this,aebImportAnFDF,"aebpro_attach.fdf") : %
1656 this.importAnFDF("aebpro_attach.fdf");\fi

```

For Version XI (or prior), there seems to be a bug. When no initial view is specified, the cover sheet should be displayed. It is but is disappears after a moment in favor of the first listed attachment in the package. To compensate for this, we force the Cover Sheet to appear after 2 seconds (trial and error value).

```

1657 \ifisPDFPackage\ifx\aeb@pdfpack@initview\@empty^^J%
1658 if(app.viewerVersion<15)^^J%
1659 var pkto=app.setTimeout("app.clearTimeOut(pkto);"
1660 +"app.execMenuItem('CollectionShowRoot');" ,5000);\fi\fi}

```

(2015/08/05) Added a \ifisPDFPackage switch.

```

1661 \newif\ifisPDFPackage\isPDFPackagefalse
1662 \newcommand{\makePDFPackage}[1]{\isPDFPackagetrue
1663   \setkeys{aebpdfpack}{viewmode=details,#1}%
1664   \ifx\aeb@pdfpack@initview\@empty
1665     \let\aeb@pdfpackage@initview\@empty
1666   \else

```

(2015/07/01) Removed \aeb@exiii and \aref(), replaced with \expandafter and argument.

```

1667     {\obeyspaces\sanitize@attach\gdef\aeb@pdfpackage@initview%
1668 {/D<\expandafter\stringiiUnicode\aeb@pdfpack@initview\relax>}}%
1669   \fi
1670   \def\aeb@PDFPackageCode{/PageMode/UseAttachments%
1671     /Collection<</Type/Collection%
1672       \aeb@pdfpackage@initview/View\aeb@pdfpack@viewmode>>}%
1673   \ifpdfmarkup
1674     \def\aeb@setPDFPackage{\literalps@out{\AEB@psMrk{Catalog}
1675       <<\aeb@PDFPackageCode>>/PUT pdfmark}}%
1676   \else\ifpdf\def\aeb@setPDFPackage{%
1677     \pdfcatalog{\aeb@PDFPackageCode}}\else
1678   \ifxetex\def\aeb@setPDFPackage{\immediate
1679     \pdfm@mark{put @catalog <<\aeb@PDFPackageCode>>}}\else
1680     \let\aeb@setPDFPackage\relax\PackageWarning{aeb_pro}
1681       {The \string\makePDFPackage/string\makePDFPortfolio\space
1682         is supported for\MessageBreak
1683         dvips, pdftex, and xetex only}}\fi\fi\fi
1684 }

```

`\makePDFPortfolio` is an alias for `\makePDFPackage`. The accepted name for a PDF Package is now a PDF Portfolio.

```
1685 \let\makePDFPortfolio\makePDFPackage
1686 \@onlypreamble\makePDFPackage

We check to see if there are any attachments, if there are, we'll import them early;
useful for \makePDFPackage and for open events that cause the view to display a
page other than the first one on opening.

1687 \let\aebp@attachments\aebp@Zero
1688 \ifx\aebp@attachsource\@empty\else\let\aebp@attachments\aebp@One\fi
1689 \ifx\aebp@attachments\@empty\else\let\aebp@attachments\aebp@One\fi
1690 \ifx\aebp@attachments\aebp@One
1691     \dev@addToDocOpen{\JS{\earlyAttachForPkgs}}%
1692 \fi

1693 % End of aebpro segment
1694 </aebpro>
1695 <*package>
1696 \AtEndOfPackage{\@nameuse{aeb@inputAttachments}}
1697 \AtBeginDocument{\@nameuse{aeb@setPDFPackage}}
1698 \@ifpackageloaded{web}{-}{
1699     {\def\aebwritelastpage{%
1700         \immediate\write\@auxout{\string\expandafter\string\gdef
1701             \string\csname\space aebLastPage\string\endcsname{\arabic{page}}}}
1702         \AtEndDocument{\clearpage\addtocounter{page}{-1}\aebwritelastpage
1703             \addtocounter{page}{1}}%
1704 }
1705 % End of package segment
1706 </package>
1707 <*unicode>
```

## 12 Support for linking to attachments.

### 12.1 `\ahyperref`, `\ahyperlink`, `\ahyperextract`

The `\ahyperref` command is for linking between parent and child, or between child and child. `\ahyperref` takes three arguments, the first optional and the second two required.

```
\ahyperref [<optargs>]{<label>}{<text>}
```

where `<label>` is one defined by either `\autolabelNum` or `\labelName`. The `<text>` is the text to be typeset around the link. The optional arguments are defined and described below.

- `goto=p2c, c2p, c2c`: The type of jump, parent to child, child to parent, and child to child. The default is `p2c`
- `page=<number>`: the page of the embedded document to jump to. Default is 0

- `view=<value>`: the view to be used for the jump. Default is `Fit`
- `dest=<string>`: jump to a named destination. When this key has a value, the values of the keys `page` and `view` are ignored.
- `open=usepref|new|existing`: open the attachment according to the user preferences, a new window, or the existing window. The default is `userpref`
- `border=visible|invisible`: Determines whether a visible rectangle appears around the link. The default is `invisible`
- `highlight=none|invert|outline|insert`: How the viewer highlights the link when the link is clicked. The default is `invert`.
- `bordercolor=r g b`: The color of the border when it is visible. The default is black.
- `linestyle=solid|dashed|underlined`: The line style of the border when it is visible. The default is `solid`
- `textcolor=<color>`: the color of the text in the hypertext link. An empty value is equivalent to `\normalcolor`.
- `linewidth=thin|medium|thick`: The line width when the border is visible. When invisible, this is set to a width of zero. The default is `thin`
- `preset=\presetCommand`: A convenience key. You can define some preset values, like so

```
\def\preseti{bordercolor={0 0 1},highlight=outline,border=visible,%
linestyle=dashed}
```

Then you can say, for example,

```
\ahyperref[dest={target1-s:intro},preset=\preseti]{target1}{Jump!}
```

**Key-value pairs of attachment links.** The following are the key-values of the optional first argument of `\ahyperref`, `\hyperlink` and `\hyperextract`.

**page** The `page` key, takes a zero-based page number.

```
1708 \define@key{ahref}{page}[0]{\def\ahref@page{#1}}
1709 \def\ahref@page{0}
```

**view** The key `view` represents an “explicit destination,” as termed by the PDF Ref, §8.2.1.

- `/XYZ left top zoom`
- `/Fit`
- `/FitH top`

- /FitV left
- /FitR left bottom right
- /FitB
- /FitBH top
- /FitBV left

For the view key, enter view=Fit or view=XYB null null null, for example. The forward slash is added later.

(2015/07) More recent advances allows \fitpage, \fitvisible, \actualsize, \fitheight, \fitwidth, and \inheritzoom to be used. These are defined in insdljs.

Each of these is preceded by a page number. The page number is entered via the page key.

If a named destination is given with the dest key, then the view and page are ignored.

```
1710 \def\ap@fslash{/}
1711 \def\@isfirstslash#1#2\@nil{\def\arg@i{#1}\ifx\ap@fslash\arg@i
1712   \else\edef\ahref@view@keyvalue{/\ahref@view@keyvalue}\fi}
1713 \define@key{ahref}{view}[/Fit]{\edef\ahref@view@keyvalue{#1}%
1714   \expandafter\@isfirstslash\ahref@view@keyvalue\@nil
1715 }
1716 \def\ahref@view@keyvalue{/Fit}
```

**dest** The dest key

```
1717 \define@key{ahref}{dest}[]{\def\ahref@dest{#1}}
1718 \let\ahref@dest\@empty
```

**open** The open key takes values of userpref, new, and existing.

```
1719 \define@choickey{ahref}{open}[\val\nr]{userpref,new,existing}
1720 {%
1721   \ifcase\nr\relax
1722     \let\ahref@open@keyvalue\@empty\or
1723     \def\ahref@open@keyvalue{/NewWindow true }\or
1724     \def\ahref@open@keyvalue{/NewWindow false }\fi
1725 }{}
1726 \let\ahref@open@keyvalue\@empty
```

**border** The border key takes values of visible and invisible.

```
1727 \define@choickey{ahref}{border}[\val\nr]{visible,invisible}
1728 {%
1729   \ifcase\nr\relax
1730     \def\ahref@border{0 0 1}%
1731     \def\ahref@invisible@border{1}\or
1732     \def\ahref@border{0 0 0}% W = 0 also?
1733     \let\ahref@invisible@border\@empty\fi
1734 }{}

```

**highlight** The highlight key takes values of none, invert, outline, and insert.

```

1735 \define@choicekey{ahref}{highlight}[\val\nr]{none,invert,outline,insert}
1736 {%
1737   \ifcase\nr\relax
1738     \let\ahref@highlight@empty\or
1739     \def\ahref@highlight{/H/I}\or
1740     \def\ahref@highlight{/H/O}\or
1741     \def\ahref@highlight{/H/P}\fi
1742 }{}

```

**bordercolor** The bordercolor key

(2017/03/08) \ahref@bordercolor can be a named color,

```

1743 \define@key{ahref}{bordercolor}[]{\def\ap@argi{#1}\ifx\ap@argi\empty
1744   \let\ahref@bordercolor\empty}else
1745   \HyColor@IfXcolor{\HyColor@XZeroOneThreeFour{#1}
1746     {\ahref@bordercolor}{}}{}
1747   {\edef\ahref@bordercolor{#1}}\fi
1748 } % PDF color
1749 \let\ahref@bordercolor\empty

```

**linestyle** The linestyle key takes values of solid, dashed, and underlined.

```

1750 \define@choicekey{ahref}{linestyle}[\val\nr]{solid,dashed,underlined}
1751 {%
1752   \ifcase\nr\relax
1753     \def\ahref@linestyle@keyvalue{/S/S}\or
1754     \def\ahref@linestyle@keyvalue{/S/D}\or % Dashed [0 0 1[3.0]]
1755     \def\ahref@linestyle@keyvalue{/S/U}\fi
1756 }
1757 \def\ahref@dashedlinestyle{/S/D}

```

**linewidth** The linewidth key takes values of thin, medium, and thick.

```

1758 \define@choicekey{ahref}{linewidth}[\val\nr]{thin,medium,thick}
1759 {%
1760   \ifcase\nr\relax
1761     \def\ahref@linewidth{1}\or
1762     \def\ahref@linewidth{2}\or
1763     \def\ahref@linewidth{3}\fi
1764 }

```

**goto** The goto key takes values of p2c, c2p, and c2c.

```

1765 \define@choicekey{ahref}{goto}[\val\nr]{p2c,c2p,c2c}
1766 {%
1767   \ifcase\nr\relax
1768     \global\let\targetDictionary\targetDictionary@pc\or
1769     \global\let\targetDictionary\targetDictionary@cp\or
1770     \global\let\targetDictionary\targetDictionary@cc\fi
1771 }{}

```

**launch** The launch key takes values of save, view, and viewnosave.

```

1772 \define@choicekey{ahref}{launch}[\val\nr]{save,view,viewnosave}
1773   {\edef\ahref@launch@type{\nr}}{}

```

```

1774 \def\ahref@launch@type{0}%
preset The preset key
1775 \define@key{ahref}{preset}[]{\def\ahref@preset{#1}}
1776 \let\ahref@preset\@empty
\ahrefcolor Color of the link text by command \ahrefcolor or as a key textcolor.
textcolor 1777 \def\ahypertxtcolor#1{\def\ahrefcolor{#1}}
1778 \def\ahrefcolor{red}
1779 \define@key{ahref}{textcolor}[]{\def\ahrefcolor{#1}}
(2015/07/01) Changed \aeb@exiii to \expandafter, changed \aref(#1) to #1
in \targetDictionary@pc and \targetDictionary@cc.
1780 \def\targetDictionary@pc#1{/T<</N<\expandafter
1781 \stringiiUnicode#1\relax>/R/C>>}
1782 \def\targetDictionary@cp#1{/T<</R/P>>}
1783 \def\targetDictionary@cc#1{%
1784 /T<</R/P/T<</N<\expandafter\stringiiUnicode#1\relax>/R/C>>>}

\factory@ahyperref@defaults The defaults, \factory@ahyperref@defaults, of \ahyperref, as indicated above
\setahyperDefaults in the descriptions of the keys. Can be redefined, I suppose. Perhaps the better
\resetahyperDefaults approach is to use \setahyperDefaults to set the defaults, possibly overwriting
the factory defaults. Use \resetahyperDefaults to cancel out the user defined
defaults. You can also use the preset key for conveniently changing the attributes
of the link.
1785 \def\factory@ahyperref@defaults{goto=p2c,page,dest,open=userpref,view,%
1786 border=invisible,highlight=invert,linestyle=solid,linewidth=thin,%
1787 bordercolor,textcolor=red,preset}
1788 \newcommand{\setahyperDefaults}[1]{\def\aeb@setahyperrefDefaults{#1}}
1789 \let\aeb@setahyperrefDefaults\@empty
1790 \newcommand{\resetahyperDefaults}{%
1791 \let\aeb@setahyperrefDefaults\@empty}

\ahyperref The \ahyperref command links a parent to its embedded children, or a child to
\ahyperlink a child. Cool. (2015/07/01) Rewrite the link to stuff, we'll use cName rather than
description to link to the attachments, this is how it should have been to begin
with. All changes dated as (2015/07/01) are the changes needed.
1792 \def\aeb@initahref#1{%
1793 \edef\expand@setkeys{%
1794 \noexpand\setkeys{ahref}{\factory@ahyperref@defaults}}%
1795 \expand@setkeys
1796 \ifx\aeb@setahyperrefDefaults\@empty\else
For the \setahyperrefDefaults we do not allow dest, goto, page, or preset.
1797 \edef\expand@setkeys{\noexpand\setkeys{ahref}%
1798 [dest,goto,page,preset]{\aeb@setahyperrefDefaults}}%
1799 \expand@setkeys
1800 \fi
1801 \setkeys{ahref}{#1}%
1802 \ifx\ahref@preset\@empty\else

```



```

1803 \edef\expand@setkeys{\noexpand
1804   \setkeys{ahref}{\ahref@preset}}\expand@setkeys\fi
1805 \let\ahref@dasharray\@empty
1806 \ifx\ahref@linestyle@keyvalue\ahref@dashedlinestyle
1807   \def\ae@border@parameters{0 0 \ahref@linewidth[3.0]}%
1808   \def\ahref@dasharray{/D[3.0]}%
1809 \else
1810   \edef\ae@border@parameters{\ahref@border}%
1811 \fi
1812 \ifx\ahref@invisible@border\@empty\def\ahref@linewidth{0}\fi
1813 \def\ahref@BS@Dict{/BS<<%
1814   \ahref@dasharray\ahref@linestyle@keyvalue/W \ahref@linewidth>>
1815 \ifx\ahref@bordercolor\@empty\else
1816   /C[\ahref@bordercolor]\fi\ahref@highlight}%
1817 }
1818 \newcommand{\ahyperref}[3] []{\begingroup
1819   \dl@preProcDefns
1820   \def\u{\string\u}%
1821   \ae@initahref{#1}%
1822   \ifx\ahref@dest\@empty
1823     \def\ahref@dest@keyvalue{%
1824       /D[\ahref@page\space\ahref@view@keyvalue]}%
1825   \else
1826     \edef\ahref@dest@keyvalue{/D(\labelRef{\ahref@dest})}%
1827   \fi
1828   \ae@ahref@setlink{#2}{#3}%
1829 }
1830 \newcommand{\ahyperlink}[3] []{\begingroup
1831   \dl@preProcDefns
1832   \def\u{\string\u}%
1833   \ae@initahref{#1}%
1834   \ifx\ahref@dest\@empty
1835     \def\ahref@dest@keyvalue{%
1836       /D[\ahref@page\space\ahref@view@keyvalue]}%
1837   \else
1838     \def\ahref@dest@keyvalue{/D(\ahref@dest)}%
1839   \fi
1840   \ae@ahref@setlink{#2}{#3}%
1841 }
1842 \def\ae@ahref@setlink#1#2{%
1843   \ifx\ahrefcolor\@empty
1844     \@eqlinktxtcolor{\let\ahypercolor\@empty\else
1845       \def\ahypercolor{\textcolor{\ahrefcolor}}\fi
1846   \setLink[\Border{\ae@border@parameters}]%
1847   \rawPDF{\ahref@BS@Dict}%
1848   \A{\ahref@dest@keyvalue/S/GoToE%
1849   \ahref@open@keyvalue\targetDictionary{#1}}{\ahypercolor{#2}}%
1850   \endgroup
1851 }

```

`\ahyperextract` The `\ahyperextract` command extracts an embedded file from the parent to simple view the document, to save and view, or simply to save. The open key determines the resulting action.

```

1852 \newcommand{\ahyperextract}[3][\begingroup\dl@preProcDefns
1853   \def\uf\string\{u}%
1854   \aeb@initahref{#1}%
1855   \ifx\ahrefcolor\@empty
1856     \@eqlinktxtcolor{\let\ahypercolor\@empty\else
1857     \def\ahypercolor{\textcolor{\ahrefcolor}}\fi
1858   \setLink[\Border{\aeb@border@parameters}]%
1859   \rawPDF{\ahref@BS@Dict}%
(2015/07/01) Removed \aref{#2}, replaced with #2
1860   \A{JS{this.exportDataObject({cName: "#2",%
1861   nLaunch: \ahref@launch@type}})}{\ahypercolor{#3}}%
1862   \endgroup
1863 }

```

### 13 `\labelName` and `\autolabelNum`

`\autolabelNum` Commands for defining the description string for the attachments. It is important  
`\autolabelNum*` that the description of the attached file, and the reference to in in the `/GoToE`  
`\labelName` exactly match. The commands define here set up the mapping between a “`LATEX`”  
 label and the description. These commands also allow the document author to  
 overwrite the default attachment descriptions. Usage:

1. `\autolabelNum[<label>]{<attachment_number>}`
2. `\autolabelNum* [<label>]{<attachment_number>}{<description>}`
3. `\labelName{<label>}{<description>}`

`\autolabelNum` and `\autolabelNum*` are used with attachments attached using the `attachments` option, while `\labelName` is used if the document author attached a file using the `docassembly` environment. `\labelName` is also used by `\autolabelNum` and `\autolabelNum*` to make definitions, in that respect, `\labelName` is the low-level command for setting up the mappings and definitions.

```

1864 \begingroup\catcode'\=\active \gdef\DQ{\string"}%
1865   \gdef\@handledblquotes{\catcode'\=\active\def"{\DQ}}%
1866   \toks0=\expandafter{\DLspecialDefs\@handledblquotes}%
1867   \xdef\DLspecialDefs{\the\toks0}%
1868 \endgroup
(2015/08/07) Add the switch \ifHandleDblQuotes switch.
1869 \newif\ifHandleDblQuotes \HandleDblQuotesfalse
1870 \def\sanitize@attach{\ifHandleDblQuotes\@handledblquotes\fi%
1871 \@makeother#\@makeother$\@makeother%\@makeother~}
1872 {\obeyspaces\gdef\getdscripCont#1{\gdef\aebp@description{#1}}%

```

```

1873 \aeb@labelName}%
1874 \gdef\getdscripStrCont#1{\gdef\aebp@description{#1}%
1875 \aeb@labelNamestar}}%

(2015/07/05) Modified \labelName and \labelName@star to gobble up any spaces
between arguments before we switch to \obeyspaces.

1876 \let\aeb@isFromAutoLabelNum\aebp@Zero
1877 \def\attachmentNamesEnv{attachmentNames}

If \labelName is outside the attachmentNames environment, use \defUniStr,
defined in forms16be.

1878 \def\labelName{\ifx\attachNamesEnv\attachmentNamesEnv
1879 \def\ap@next{\labelName@Env}\else\def\ap@next{\defUniStr}\fi\ap@next}
1880 \def\labelName@Env#1{\bgroup\let\aeb@isFromAutoLabelNum\aebp@Zero %dps23
1881 \@for\x:=ABCDEFabcdef\do{\@makeother\x}%
1882 \def\aebp@labelname{#1}\@ifnextchar\bgroup{\labelName@i}{}}

(2015/08/02) \autoLabelNum used to call \labelName, now it calls \labelName@a,
which sets a switch to notify \aeb@labelName of the origins of the call.

1883 \def\labelName@a#1{\bgroup\let\aeb@isFromAutoLabelNum\aebp@One
1884 \def\aebp@labelname{#1}\@ifnextchar\bgroup{\labelName@i}{}}
1885 \def\labelName@i{\obeyspaces\sanitize@attach\getdscripStrCont}%
1886 \def\labelName@star#1{\bgroup\@for\x:=ABCDEFabcdef\do{\@makeother\x}%
1887 \def\aebp@labelname{#1}\@ifnextchar\bgroup{\labelName@star@i}{}}%
1888 \def\labelName@star@i{\obeyspaces\sanitize@attach\getdscripStrCont}%
1889 \newcommand{\autolabelNum}{%
1890 \ifstar{\aeb@autoLabelNumstar}{\aeb@autoLabelNum}}
1891 \newcommand\aeb@autoLabelNum[2][\def\alnargi{#1}%
1892 \def\@thisattachnum{#2}%
1893 \ifx\alnargi\@empty
1894 \def\aeb@next{%
1895 \labelName@a{attach#2}{AeB\space Attachment\space #2}}%
1896 \else
1897 \def\aeb@next{%
1898 \labelName@a{#1}{AeB\space Attachment\space #2}}%
1899 \fi
1900 \aeb@next%
1901 }
1902 \newcommand{\aeb@autoLabelNumstar}[2][\def
1903 \alnargi{#1}\gdef\@thisattachnum{#2}%
1904 \ifx\alnargi\@empty\def\aeb@next{\labelName@star{attach#2}}\else
1905 \def\aeb@next{\labelName@star{#1}}\fi
1906 \aeb@next%
1907 }
1908 \def\aeb@labelName{%

(2015/08/02) The definitions differ depending on whether it comes from the com-
mand \autoLabelNum or from \labelNum.

1909 \if\aeb@isFromAutoLabelNum\aebp@One
1910 \global\csarg\let
1911 {aeb@aDescript@attach\@thisattachnum}\aebp@description

```

```

1912     \global\csarg\let
1913         {aeb@aNam@attach\@thisattachnum}\aebp@labelname
1914     \else
1915         \global\csarg\let
1916             {aeb@aDescript@aebp@labelname}\aebp@description
1917         \global\csarg\let
1918             {aeb@aNam@aebp@labelname}\aebp@labelname\fi\egroup}
1919 \def\aeb@labelNamestar{%
1920     \global\csarg\let
1921         {aeb@aDescript@attach\@thisattachnum}\aebp@description
1922     \global\csarg\let{aeb@aNam@attach\@thisattachnum}\aebp@labelname
1923     \global\csarg\let{aeb@aDescript@aebp@labelname}\aebp@description
1924     \global\csarg\let{aeb@aNam@aebp@labelname}\aebp@labelname\egroup}

```

We need a general verbatim write environment, here is the one used by `exerquiz` and several other packages. If `exerquiz` is loaded, no need to redefine it.

```

1925 \@ifpackageloaded{exerquiz}{-}
1926 {%-
1927     \def\verbatimwrite{\@bsphack
1928         \let\do\@makeother\dospecials
1929         \catcode'\^M\active\catcode'\^^I=12%
1930         \def\verbatim@processline{%
1931             \immediate\write\verbatim@out
1932             {\the\verbatim@line}}%
1933         \verbatim@start}
1934     \def\endverbatimwrite{\@esphack}
1935 }-

```

**attachmentNames** Enclose the `\labelName` and `\autoLabelNum` commands this environment, which writes its contents verbatim to the file `\jobname_xref.cut` and inputs it back in. Child files can input this file to get the labels assigned for them by the parent document.

```

1936 \let\attachNamesEnv\@empty
1937 \newenvironment{attachmentNames}{%-
1938     \newwrite \aeb@attachments@xref
1939     \immediate\openout \aeb@attachments@xref \jobname_xref.cut
1940     \let\verbatim@out\aeb@attachments@xref
    (2015/08/07) Set \ifHandleDblQuotes to true, added a group.
1941     \immediate\write\verbatim@out{\string\begingroup
1942         \string\HandleDblQuotestrue^^J%
1943         \string\def\string\attachNamesEnv{attachmentNames}}%
1944     \verbatimwrite}{%-
1945     \endverbatimwrite
1946     \immediate\write\verbatim@out{\string\endgroup}}%
1947     \immediate\closeout \aeb@attachments@xref
1948     \aftergroup\aeb@Input@xref}
1949 \def\aeb@Input@xref{\InputIfFileExists{\jobname_xref.cut}{-}{-}}
1950 % End of unicode segment

```

```

1951 </unicode>
(2013/04/01) The createicon block separates out the import icon macros, doc
and pre-docassembly stuff.
1952 % Begin aebpro and createicon segments
1953 <*aebpro | createicon>

```

## 14 Document Assembly Methods

Special “helper” commands and one environment are defined to take advantage of Acrobat’s extensive library of security restricted methods. It is assumed the document author has properly installed `aeb_pro.js`.

### 14.1 The docassembly environment

`docassembly` This is a wrapper environment for the `execJS` environment of the `insDLJS` Package. Place JavaScript lines in this environment and the script will execute one time after the PDF has been created and opened in Acrobat Pro for the first time.

```

1954 \newenvironment{docassembly}{%
1955 \execJS{docassembly}
1956 }{\endexecJS}

```

### 14.2 Supported Assembly JS API

These are convenience commands – called JavaScript helper commands – to executing security restricted JavaScript. The JS methods are defined in the `aeb_pro.js` file, kept as folder JavaScript. These commands are executed in a verbatim environment where ‘\’ is still the escape character. Each of the JavaScript helper commands expects a left parenthesis ‘(’ following the command name *on the same line* as the command name.<sup>1</sup> See the example below for correct usage.

```

\begin{docassembly}
\addWatermarkFromFile({
  bOnTop:false,
  cDIPath:"/C/AcroTeX/AcroPackages/ManualBGs/Manual_BG_Print_AeB.pdf"
});
\end{docassembly}

```

For each of the methods below, see the *JavaScript for Acrobat API Reference*.

The command `\theDocObject` is normally set to `this`, meaning the current document. You may need to set it to some other doc object if you are trying to access a doc object other than the current one.

Revised all these function that use `\theDocObject`; the revisions allow you to change the use of `\theDocObject`. The following are support commands for changing `\theDocObject` from within the `docassembly` environment.

<sup>1</sup>This requirement is consistent with JavaScript function usage.

`\chgDocObjectTo` All the JavaScript helper commands use `\theDocObject`, which is defined to be the `this` object. To change it within the `docassembly` environment is difficult. The next command aids in that problem.

```
1957 \let\ap@mrk\@empty
1958 \def\ap@gobtocomma#1,{
1959 \providecommand\chgDocObjectTo[2]{%
1960     \def#1##1\ap@mrk{#2,\ap@gobtocomma##1}}
```

The above defines a new command given by `#1`. The command has one argument which is all content up to the terminating mark `\ap@mrk`. The trick to removing `\thisDocObject` and replacing it with `#2`, in the above definition, we insert `'(#2'` followed by `\ap@gobtocomma`, which absorbs `\thisDocObject`, (absorbs everything through the first comma), followed by all content (`##1`); the second `\@gobble` absorbs the left parenthesis that opens the argument.

```
1961 \def\ap@TF{aebTrustedFunctions}
```

An example of usage of `\chgDocObject` is `\chgDocObjectTo{\newDO}{doc}` expanded above the `docassembly` environment. Later, we can say,

```
\chgDocObjectTo{\newDO}{doc}
\begin{docassembly}
...
\docSaveAs\newDO({ cPath: _path });
...
\end{docassembly}
```

That is, it is placed immediately after any of the commands below that uses `\theDocObject`.

`\theDocObject` This command is used in the definition of all JavaScript helper commands, as seen in the definition of `\DeclareJSHelper` below. It is set to the doc object `this`. It can be changed using `\chgDocObjectTo`, as described above.

```
1962 \def\theDocObject{this}
```

`\DeclareJSHelper` A general purpose command for defining what I am calling JavaScript helper commands.

```
1963 \providecommand\DeclareJSHelper[2]{%
1964     \def#1##1({\ap@TF{##1\theDocObject,#2,\ap@mrk}}
```

For example, we declare `\DeclareJSHelper{\docSaveAs}{aebDocSaveAs}` below, the declaration defines a new command, `\docSaveAs`:

```
\def\docSaveAs#1({\ap@TF{#1\theDocObject,aebDocSaveAs,\ap@mrk}}
```

Note that the argument of `\docSaveAs` is delimited by the left parenthesis, thus `#1` is everything through that opening parenthesis. This approach allows more flexibility in the definition, there can be spaces following the command name `\docSaveAs` (`{path: _path}`), for example.

`\retnAbsPathAs(js-var)` Several methods require an absolute path to the current folder. The code is,

```
var _path=this.path;
var pos=_path.lastIndexOf("/");
_path=_path.substring(0,pos);
```

We simplify this code for the document author in the form of the command `\retnAbsPathAs(js-var)`, where *js-var* is a JavaScript variable that will hold the absolute path to the current folder; eg, `\retnAbsPathAs(_path)`; expands to the above code.

```
1965 \def\retnAbsPathAs(#1){%
1966   var #1=this.path;^^J%
1967   var pos=#1.lastIndexOf("/");^^J%
1968   #1=#1.substring(0,pos)}
```

Notice the argument *js-var* is delimited by *parentheses*, not braces.

```
1969 % End aebpro and createicon segments
1970 </aebpro | createicon>
1971 <*aebpro>
```

`\addWatermarkFromFile` This is the method `Doc.addWatermarkFromFile`.  
1972 \DeclareJSHelper{\addWatermarkFromFile}{aebAddWatermarkFromFile}

`\addWatermarkFromText` This is the method `Doc.addWatermarkFromText`.  
1973 \DeclareJSHelper{\addWatermarkFromText}{aebAddWatermarkFromText}

```
1974 % End aebpro segment
1975 </aebpro>
1976 <*aebpro | createicon>
```

`\importIcon` This is the method `Doc.importIcon`.  
1977 \DeclareJSHelper{\importIcon}{aebImportIcon}

```
1978 % End aebpro and createicon segments
1979 </aebpro | createicon>
1980 <*aebpro>
```

`\importSound` This is the method `Doc.importSound`.  
1981 \DeclareJSHelper{\importSound}{aebImportSound}

`\appopenDoc` This is the method `app.openDoc`.  
1982 \DeclareJSHelper{\appopenDoc}{aebAppOpenDoc}

This is the method `Doc.saveAs`.

`\docSaveAs` This is the document method `Doc.saveAs`.  
1983 \DeclareJSHelper{\docSaveAs}{aebDocSaveAs}

`\browseForDoc` Access to using the method `app.browseForDoc()`, for example,  

```
var oRetn=\browseForDoc({bSave:true,cFilenameInit: "myNewDoc.pdf"});
```

See the Acrobat JavaScript API reference for details on what is returned.  
1984 `\DeclareJSHelper{\browseForDoc}{aebBrowseForDoc}`

`\insertPages` This is the method `Doc.insertPages`.  
1985 `\DeclareJSHelper{\insertPages}{aebInsertPages}`

`\createTemplate` This is the method `Doc.createTemplate`.  
1986 `\DeclareJSHelper{\createTemplate}{aebCreateTemplate}`

`\importDataObject` This is the method `Doc.importDataObject`. `\attachFile` is a more intuitive  
`\attachFile` name for the operation performed.  
1987 `\DeclareJSHelper{\importDataObject}{aebImportDataObject}`  
1988 `\DeclareJSHelper{\attachFile}{aebImportDataObject}`

`\executeSave` To save the document, use at the *end of the doc assembly environment*. Usage:  
`\executeSave()`. The `\@gobble` used below absorbs the comma that is placed  
immediately after the second argument by `\DeclareJSHelper`.  
1989 `\DeclareJSHelper{\executeSave}{aebSaveAs,"Save"@gobble}`

`\extractPages` API to extract pages from a newly formed document.  
1990 `\DeclareJSHelper{\extractPages}{aebExtractPages}`

`\mailDoc` Used by eEnvelope to programmatically mail a document.  
1991 `\DeclareJSHelper{\mailDoc}{aebMailDoc}`

`\sigInfo` The `\sigInfo` command is used for entering signing formation into what will  
`\signatureSign` become an object. `\signatureSign` takes no arguments, but uses the info entered  
`\signatureSetSeedValue` by `\sigInfo`. An example is

```
\begin{docassembly}
\sigInfo{
  cSigFieldName: "sigOfDPS", ohandler: security.PPKLiteHandler,
  cert: "D_P_Story.pfx", password: "dps017",
  oInfo: { location: "Niceville, FL",
    reason: "I am approving this document",
    contactInfo: "dpstory@acrotex.net",
    appearance: "My Signature" }
};
\signatureSign
\end{docassembly}
```

The `\sigInfo` command is a latex interface to creating the `oSigInfo` object.  
1992 `\newcommand{\sigInfo}{var oSigInfo=}`  
1993 `\def\sigFieldObj(#1){var oSigField=this.getField(#1)}`



For the `\signatureSetSeedValue`, the field object is required. This function assumes that the JavaScript variable `oSigField` is the field object. For example,

```
\begin{docassembly}
\sigFieldObj("sigOfDPS");
\signatureSetSeedValue({
  lockDocument:true,
  appearanceFilter:"My Signature",
  reasons: ["This is a reason", "This is a better reason"],
  flags:0x80|0x100|0x8
});
\end{docassembly}
```

The `signatureSetSeedValue()` method seeds a signature field with various default values available to the signer.

```
\begin{docassembly}
var sv={
  mdp: "defaultAndComments",
  reasons: ["This is a reason", "This is a better reason"],
  flags:0x80|0x100|0x8
};
\sigFieldObj("sigOfDPS");
\signatureSetSeedValue(sv);
\end{docassembly}
```

```
1994 \def\signatureSetSeedValue#1{%
1995   \ap@TF( oSigField, aebSignatureSetSeedValue, }
```

The function `\signatureSign` takes the info in the `oSigInfo` object, gets the security handler object, logs into the handler, calls `signatureSetSeedValue` if the `sv` property is in the `oSigInfo` object, and signs the field.

```
1996 \begin{defineJS}[\makecmt%\dfnJSCR{^J}]{\signatureSign}
1997 if ( typeof oSigInfo.oHandler=="undefined" )
1998   oSigInfo.oHandler=security.PPKLiteHandler;
1999 var engine=aebTrustedFunctions( security,%
2000 aebSecurityGetHandler, oSigInfo.oHandler );
2001 var path2Cert = (typeof oSigInfo.path2Cert == "undefined") ? %
2002 aebTrustedFunctions( this, aebAppGetPath,%
2003 {cCategory:"user"} )+"/Security+"/"+oSigInfo.cert : %
2004 oSigInfo.path2Cert;
2005 aebTrustedFunctions( engine, aebSecurityHandlerLogin,%
2006 { cPassword: oSigInfo.password, cDIPath: path2Cert});
2007 var oSigField = this.getField(oSigInfo.cSigFieldName);
2008 oSigInfo.oInfo.password=oSigInfo.password;
2009 if ( typeof oSigInfo.sv!="undefined" ) {
2010   for (var o in oSigInfo.sv )
2011     oSigInfo.oInfo[o]=oSigInfo.sv[o];
2012 }
2013 var oSigArgs={ oSig: engine, oInfo: oSigInfo.oInfo };
2014 if ( typeof oSigInfo.cLegalAttest!="undefined" )
```

```

2015 oSigArgs.cLegalAttest=oSigInfo.cLegalAttest;
2016 if ( typeof oSigInfo.cDIPath!="undefined")
2017 oSigArgs.cDIPath=oSigInfo.cDIPath;
2018 if ( typeof oSigInfo.bUI!="undefined")
2019 oSigArgs.bUI=oSigInfo.bUI;
2020 aebTrustedFunctions( oSigField, aebSignatureSign, oSigArgs );
2021 \end{defineJS}

```

`\certifyInvisibleSign` This command uses the trusted version of `certifyInvisibleSign` to sign. The command requires that `\sigInfo` is populated appropriately.

```

\begin{docassembly}
\sigInfo{
  cert: "DonStory.pfx", password: "myPassword",
  cLegalAttest: "Certified using JavaScript",
  bUI:false,
  oInfo: {
    location: "Niceville, FL",
    reason: "I am certifying this document",
    mdp: "defaultAndComments",
  }
};
\certifyInvisibleSign
\end{docassembly}

```

```

2022 \begin{defineJS}[\makecmt\%\dfnJSCR{^^J}]{\certifyInvisibleSign}
2023 if ( typeof oSigInfo.oHandler=="undefined" )
2024 oSigInfo.oHandler=security.PPKLiteHandler;
2025 var engine=aebTrustedFunctions( security, %
2026 aebSecurityGetHandler, oSigInfo.oHandler );
2027 var path2Cert=aebTrustedFunctions( this, aebAppGetPath, %
2028 {cCategory:"user"} )+"/Security+"/"+oSigInfo.cert;
2029 aebTrustedFunctions( engine, aebSecurityHandlerLogin, %
2030 { cPassword: oSigInfo.password, cDIPath: path2Cert});
2031 oSigInfo.oInfo.password=oSigInfo.password;
2032 var oSigArgs={ oSig: engine, oInfo: oSigInfo.oInfo };
2033 if ( typeof oSigInfo.cLegalAttest!="undefined" )
2034 oSigArgs.cLegalAttest=oSigInfo.cLegalAttest;
2035 if ( typeof oSigInfo.cDIPath!="undefined")
2036 oSigArgs.cDIPath=oSigInfo.cDIPath;
2037 if ( typeof oSigInfo.bUI!="undefined")
2038 oSigArgs.bUI=oSigInfo.bUI;
2039 aebTrustedFunctions( this, aebCertifyInvisibleSign, oSigArgs );
2040 \end{defineJS}

```

```

2041 % End aebpro segment
2042 </aebpro>
2043 <*aebpro | createicon>

```

## 14.3 Pre-docassembly Commands

In this section, we'll gather some “useful” commands for combining several `docassembly` tasks together. The `docassembly` environment is a partial-verbatim environment, expansion is severely limited. (`\` is the escape sequence, but `{}` are no longer grouping characters; hence macros with no argument can be expanded.) The trick is to expand before placing the lines in the `docassembly` environment.

### 14.3.1 Import and Place Images

`\declareImageAndPlacement` With this command, the document author can import images (icons) into the document give them names, and associate the image with a push button, specifically, once created by the `\placeImage` command, defined below. The command may be used several times, once for each image (icon) that is to be embedded and placed.

This command should appear outside the `docassembly` environment.

In the example below, give an image a name `manAvatar`, specify the path (`man1.pdf`) and list the names of the push buttons (ones created by `\placeImage`) where this image is to appear as a button face.

```
\declareImageAndPlacement
{
  name=manAvatar,path=man1.pdf,placement={Avatar1,Avatar2}
}
```

The macro `\@importIconToks` will contain all JavaScript the commands created by the command `\declareImageAndPlacement`. `\aebp@image@cnt` is used for automatically generating a name if one is not given.

```
2044 \let\predocassemJS\@empty
2045 \def\aebp@image@cnt{0}
```

These keys are valid for the argument of `\declareImageAndPlacement`: `name`, `path`, and `placement` are required; if the image is a PDF, `page=0` is the default, otherwise, you can specify from which page the PDF image is to be extracted from the PDF file.

```
name 2046 \define@key{importIcons}{name}[]{\def\importIcons@name{#1}}
path 2047 \define@key{importIcons}{path}[]{\def\importIcons@path{#1}}
page 2048 \define@key{importIcons}{page}[]{\def\importIcons@page{#1}}
placement 2049 \define@key{importIcons}{placement}[]{\def\importIcons@placement{#1}}
embedonly 2050 \define@boolkey{importIcons}{embedonly}[true]{}
firstpage 2051 \define@key{importIcons}{firstpage}[]{\def\importIcons@firstpage{#1}}
lastpage 2052 \define@key{importIcons}{lastpage}[]{\def\importIcons@lastpage{#1}}
```

The next two macros parse `\importIcons@placement` after it has been broken down into its comma-delimited parts. We pick off the optional argument using `\aebp@getFace`, then we call `\aebp@getPlacement` to get the field name. These are executed from within `\declareImageAndPlacement` below.

If there is no optional argument, the default is 0, for the normal appearance state.

```
2053 \newcommand{\aebp@getFace}[1][0]{%
```

```

2054 \def\importIcons@face{#1}\aebp@getPlacement}
2055 \def\aebp@getPlacement#1\@nil{\def\importIcons@fieldname{#1}%
2056 \expandafter\ap@strpspcs\importIcons@fieldname\@nil
2057 \edef\importIcons@fieldname{\to@nilarg}%
2058 }

```

The command `\declareImageAndPlacement` takes one argument, key-value pairs as described above. This command may be use many times, once for each image.

`\declareMultiImages` Here is another macro, created after `\declareImageAndPlacement`, that enables the declaration of multiple images and placements. The following is an example of usage:

```

\declareMultiImages{%
  {name=manAvatar,path=man1.pdf,placement={Avatar1,Avatar2}}
  {path=girl.png,placement={Avatar3,Avatar4}}
  {path=AcroFord.jpg,placement=AcroFord}
  {path=scot.gif,placement=Scot}
}

```

(9/24/2009) Now you can enter the face number as an optional argument in the placement value: [0] (default, normal icon), [1] (down icon) [2] (rollover icon). The optional argument precedes the field name. For Example

```

\declareMultiImages
{%
  {path=graphics/man1.pdf,placement={Avatar1,Avatar2}}
  {path=graphics/girl.png,placement={ [2] Avatar1, [2] Avatar2}}
}

```

```

2059 \newcommand{\declareMultiImages}[1]{%
2060 \edef\@tforExp{\noexpand\@tfor\noexpand\@dipArgs:=#1}%
2061 \@tforExp\do{%
2062 \expandafter\declareImageAndPlacement
2063 \expandafter{\@dipArgs}%
2064 }%
2065 }

```

The definition of `\declareImageAndPlacement`. Keys recognized are defined above: name, path, page, placement.

```

2066 \newcommand{\declareImageAndPlacement}[1]{%
2067 \edef\@tempexp{\noexpand\setkeys{importIcons}{name,path,page,%
2068 placement,embedonly=false,#1}}\@tempexp
2069 \ifx\importIcons@name\@empty{\count0=\ae@image@cnt
2070 \advance\count0by1\relax\edef\ae@image@cnt{\the\count0}}%
2071 \edef\importIcons@name{aeImage\ae@image@cnt}\fi
2072 \ifx\importIcons@path\@empty
2073 \PackageError{aeb_pro}{The 'path' key is required}
2074 {Please give the path of the icon.}\fi
2075 \ifx\importIcons@placement\@empty\ifKV@importIcons@embedonly\else
2076 \PackageWarningNoLine{aeb_pro}{%
2077 The 'placement' key was not found for the\MessageBreak

```

```

2078     image named "\importIcons@name".\MessageBreak
2079     Specify this key, or insert the image manually in\MessageBreak
2080     the docassembly environment.}\fi\fi
2081 \xdef\predocassemJS{\predocassemJS
2082     var retn=\importIcon({cName:"\importIcons@name",%
2083     cDIPath:"\importIcons@path"%
2084     \ifx\importIcons@page\@empty\else,%
2085     nPage:\importIcons@page\fi});^^J%
2086     if ( retn==-1 ) console.println("The file
2087     \string"\importIcons@path\string\" could not be opened.");^^J%
2088     if ( retn==-2 ) console.println("Selected page
2089     (page=\importIcons@page) for \string"\importIcons@path\string\"
2090     is invalid.");^^J}%
2091 \ifx\importIcons@placement\@empty\else
2092     \@for\@fieldname:=\importIcons@placement\do{%
2093     \expandafter\aebp@getFace\@fieldname\@nil
2094     \xdef\predocassemJS{\predocassemJS
2095     var f = this.getField("\importIcons@fieldname");^^J%
2096     if ( f == null ) console.println("The field
2097     \string"\importIcons@fieldname\string\"
2098     does not exist.");^^J%
2099     try { f.buttonSetIcon(this.getIcon("\importIcons@name"),%
2100     \importIcons@face); }
2101     catch(e){};^^J}%
2102     }%
2103 \fi
2104 }
2105 \@onlypreamble\declareImageAndPlacement

```

### 14.3.2 Embed Multi-page Images

`\iconNameI` `\iconNameI` is the “template name” for the icons, this template is a JavaScript string and expands to a function of `i`. The default definition is given below.

```

2106 \newcommand{\iconNameI}[1]{\def\IconNameI{##1}{#1}}
2107 \iconNameI{#1+"."+i}
2108 %\iconNameI{#1+i}
2109 \@onlypreamble\iconNameI

```

`\embedMultiPageImages` `\embedMultiPageImages` is designed for a PDF filled with icon pages. It sets up a JavaScript loop, and imports the specified page range. The name of the icon is determined by the command `\iconNameI`, this is function of the JavaScript variable `i`, the index for the loop. The command embeds the icons only. The parameters are...

key-value pairs: `firstpage`, `lastpage`, `placement`, `name` and `path` are the only keys recognized.

```

2110 \newcommand{\embedMultiPageImages}[1]{%
2111     \setkeys{importIcons}{name,path,page,placement,%
2112     firstpage,lastpage,#1}%

```

```

2113 \ifx\importIcons@firstpage@empty
2114 \def\importIcons@firstpage{1}\else
2115 {\count0=\importIcons@firstpage}\fi
2116 \ifx\importIcons@lastpage@empty
2117 \def\importIcons@lastpage{this.numPages}\else
2118 {\count0=\importIcons@lastpage}\fi
2119 \ifx\importIcons@name@empty
2120 \PackageError{aeb_pro}{The 'name' key is required}
2121 {Please give the icon family a name.}\fi
2122 \ifx\importIcons@path@empty
2123 \PackageError{aeb_pro}{The 'path' key is required}
2124 {Please give the path of the icon.}\fi
2125 \def\r{^~J}\def\t{\space\space\space\space}%
2126 \let\predocassemJSPlace@empty
2127 \ifx\importIcons@placement@empty\else
2128 \@for\@fieldname:=\importIcons@placement\do{%
2129 \expandafter\@ebp@getFace\@fieldname@nil
2130 \edef\predocassemJSPlace{\predocassemJSPlace\t
2131 var f = this.getField("\importIcons@fieldname."+i);\r\t
2132 if ( f == null ) console.println("The field
2133 \string"\importIcons@fieldname."+i+"\string\"
2134 does not exist.");\r\t
2135 try { f.buttonSetIcon(this.getIcon(
2136 \IconNameI("\importIcons@name")),,%
2137 \importIcons@face); }
2138 catch(e){};\r}%
2139 }%
2140 \fi
2141 \xdef\predocassemJS{\predocassemJS
2142 var l=\importIcons@lastpage-\importIcons@firstpage+1;\r
2143 for (var i=0; i<l; i++) {\r\t
2144 var retn=\importIcon({%
2145 cName:\IconNameI("\importIcons@name"),%
2146 cDIPath:\importIcons@path",%
2147 nPage:\importIcons@firstpage+i-1});\r\t
2148 if ( retn==-1 ) console.println("The file
2149 \string"\importIcons@path\string\" could not
2150 be opened.");\r\t
2151 if ( retn==-2 ) console.println("Selected page
2152 (page="+i+1) for \string"\importIcons@path\string\"
2153 is invalid.");\r
2154 \predocassemJSPlace
2155 }\r
2156 var oIcon=this.getIcon("nullIcon");\r
2157 if ( oIcon == null )
2158 this.addIcon("nullIcon", this.createIcon("", 0, 0));\r
2159 }%
2160 }
2161 \@onlypreamble\embedMultiPageImages
2162 % End aebpro and createicon segments

```

```

2163 </aebpro | createicon>
2164 <*aebpro>
2165 % End aebpro segment
2166 </aebpro>
2167 <*animecommon>

```

`\placeAnimeCtrlBtnFaces` This command is part of the button anime set. It embeds (using the command `\declareMultiImages`) the icon appearances of the anime controls, The field names for these controls have a name of the form

```
<anime_basename>4BtnCtrl.<fieldName>
```

where `<anime_basename>` is the base name of the field that displays the images; and `<fieldName>` is the name of the particular control.

The first parameter is the relative or absolute path to the icons set (this path is used by Acrobat JavaScript, not by L<sup>A</sup>T<sub>E</sub>X; the second parameter is a comma-delimited list of button animes to be controlled. An example follows:

```

\placeAnimeCtrlBtnFaces{graphics/btn_anime_icons1.pdf}%
  {myAnimation,mysine,FamPhotos}
2168 \newcommand{\placeAnimeCtrlBtnFaces}[3][\pathToBtnCtrlIcons]{\begingroup
2169   \toks@={\aep@I{\aep@setAFirst}{animeFirst}}%
2170           \aep@I{\aep@setASB}{animeSB}}%
2171           \aep@I{\aep@setAPB}{animePB}}%
2172           \aep@I{\aep@setAPaus}{animePause}}%
2173           \aep@I{\aep@setAPF}{animePF}}%
2174           \aep@I{\aep@setASF}{animeSF}}%
2175           \aep@I{\aep@setALast}{animeLast}}%
2176           \aep@I{\aep@setAPlus}{animePlus}}%
2177           \aep@I{\aep@setAMinus}{animeMinus}}%
2178   \def\aep@I##1##2{\let##1\@empty}\the\toks@
2179   \def\aep@I##1##2{\edef##1{##1,\aep@thisAnime4BtnCtrl.##2}}%
2180   \@for\aep@thisAnime:=#3\do{\the\toks@}%
2181   \def\aep@I##1##2{\edef##1{\expandafter\@gobble##1}}\the\toks@
2182   \ifx\pathToBtnCtrlIcons\@empty
2183     \let\@pathsep\@empty\else\def\@pathsep{/}\fi
2184   \declareMultiImages{

```

A document author may supply his own icon set, they must be placed in a PDF document in the expected order:

Page	Icon
0	Go to first frame
1	Go to last frame
2	Step back one frame
3	Step forward one frame
4	Play backward
5	Play forward
6	Pause
7	Increase speed
8	Decrease speed

```

2185     {path=#1\@pathsep#2,page=0,placement={\aep@setAFirst}}
2186     {path=#1\@pathsep#2,page=2,placement={\aep@setASB}}
2187     {path=#1\@pathsep#2,page=4,placement={\aep@setAPB}}
2188     {path=#1\@pathsep#2,page=6,placement={\aep@setAPaus}}
2189     {path=#1\@pathsep#2,page=5,placement={\aep@setAPF}}
2190     {path=#1\@pathsep#2,page=3,placement={\aep@setASF}}
2191     {path=#1\@pathsep#2,page=1,placement={\aep@setALast}}
2192     {path=#1\@pathsep#2,page=7,placement={\aep@setAPlus}}
2193     {path=#1\@pathsep#2,page=8,placement={\aep@setAMinus}}
2194   }%
2195   \endgroup
2196 }
2197 \@onlypreamble\placeAnimeCtrlBtnFaces

2198 % End animecommon segment
2199 </animecommon>
2200 <*aebpro>

```

`\placeImageToBtn` `\iconNameI` is the “template name” for the icons, this template is a JavaScript string and expands to a function of `i`. The default definition is given below.

```

2201 \newcommand{\placeImageToBtn}[1]{%
2202   \setkeys{importIcons}{name,path,page,placement,#1}%
2203   \ifx\importIcons@placement\@empty\else
2204     \@for\@fieldname:=\importIcons@placement\do{%
2205       \expandafter\aebp@getFace@\@fieldname\@nil
2206       \xdef\predocassemJS{\predocassemJS
2207         var f = this.getField("\importIcons@fieldname");^^J%
2208         if ( f == null ) console.println("The field
2209           \string"\importIcons@fieldname\string\"
2210             does not exist.");^^J%
2211         try { f.buttonSetIcon(this.getIcon("\importIcons@name"),%
2212           \importIcons@face); }
2213         catch(e){};^^J}%
2214     }%
2215   \fi
2216 }

```

### 14.3.3 `\insertPreDocAssembly`

`\insertPreDocAssembly` This command is to be used within the `docassembly` environment, and expands to all the JavaScript lines created by `\declareImageAndPlacement`.

```

2217 % End aebpro segment
2218 </aebpro>
2219 <*aebpro | createicon>
2220 \newcommand{\insertPreDocAssembly}{\predocassemJS}
2221 \let\importAndSetImages\insertPreDocAssembly

```



## 14.4 Placing an Image (`\placeImage`)

`\placeImage` This command creates the target push buttons for the images imported using `\declareImageAndPlacement` and `\importAndSetImages`. It creates a readonly push button with an icon only appearance. The first argument can be used to modify the appearance (as described in the eforms manual), the second is the name of the field (this is referenced in the `placement` key of `\declareImageAndPlacement`, the third and fourth parameters are the width and height, respectively of the push button.

```
2222 \newcommand{\placeImage}[4][\pushButton[  
2223   \BC{\BG{\S{S}\Ff{\FfReadOnly}\autoCenter{n}\FB{true}  
2224   #1\I{null}\TP{1}]{#2}{#3}{#4}}  
  
2225 % End of aebpro and createicon segments  
2226 </aebpro | createicon)  
2227 <*animecommon)
```

## 14.5 Button and Ocg Animation (Anime)

The following are the key-value pairs common to both button and ocg anime.

`\animeSetup` `\animeSetup` can be used to set the anime parameters when `controls=none`, and  
`\widthFirstRow` you want to move button set to a place other than the default location, which is below the anime. After setting the parameters, we have some code for calculating the `\widthFirstRow`. this value is used by `\insertCtrlButtons` to set the width of an enclosing `\parbox`, the first row is always wider than the second.

```
2228 \newcommand{\animeSetup}[1]{%  
2229   \let\animeSetupPresets\@empty  
2230   \edef\@tmpExp{\noexpand\setkeys{btnAnime}{#1}}\@tmpExp  
2231 %   \setkeys{btnAnime}{#1}%  
2232   \ifx\btnAnimeCtrlBdryColor\@empty\else
```

Calculate the presets for the buttons as passed by the key-values of `\animeSetup`.

```
2233   \edef\@tmpExp{\noexpand\g@addto@macro%  
2234     \noexpand\animeSetupPresets{%  
2235       \noexpand\BC{\btnAnimeCtrlBdryColor}}}\@tmpExp  
2236   \edef\@tmpExp{\noexpand\g@addto@macro%  
2237     \noexpand\animeSetupPresets{%  
2238       \noexpand\W{\btnAnimeCtrlBdryWidth}}}\@tmpExp\fi
```

Calculate the value of `\@btnanimerowsep`, the distance between rows

```
2239   \edef\@btnanimerowsep{\btnanimerowsep}%  
2240 %   \setlength{\@tempdima}{\btnanimerowsep+2bp}%  
2241 %   \edef\@btnanimerowsep{\the\@tempdima}%
```

Calculate the width of the widest row, usually the first.

```
2242   \setlength{\@tempdima}{%  
2243     (\btnAnimeCtrlW+2bp)*\numWidgetsFirstRow+%  
2244     (\btnanimebtnsep*(\numWidgetsFirstRow-1))}%  
2245   \addtolength{\@tempdima}{\addSpaceBtwnPMBtns}%
```

```

2246 \edef\widthFirstRow{\the\@tempdima}%
2247 }

```

`\insertCtrlButtons` This command can be used to place the anime controls somewhere other than at the default location (under the anime).

```

2248 \newcommand{\insertCtrlButtons}[1][c]{%
2249 \parbox[#1][\totalheight+2bp]{\widthFirstRow}{%
2250 \offinterlineskip\centering\ae@btn@anime@controls}}
2251 \newcommand{\setspaceBtwnPMBtns}[1]{%
2252 \settowidth{\@tempdima}{#1\kern2bp}%
2253 \xdef\tmp@spaceBtwnPMBtns{\the\@tempdima}}%
2254 \edef\@spaceBtwnPMBtns{\tmp@spaceBtwnPMBtns}
2255 \setspaceBtwnPMBtns{\space}
2256 \def\addSpaceBtwnPMBtns{\ifKV@btnAnime@usetworows0bp\else
2257 \ifKV@btnAnime@nospeedcontrol0bp\else\@spaceBtwnPMBtns\fi\fi}

```

**Define Key-value Pairs for anime.** The next set of code defines the key-value

```

type pairs for \btnAnime and ocgAnime.
fieldName 2258 \define@choicekey+{btnAnime}{type}[\val\nr]%
ocgAnimeName 2259 {loop,palindrome,stopatboundary}[stopatboundary]{%
iconName 2260 \edef\ae@bp@btnAnime@type{\nr}}{}
speed 2261 \define@key{btnAnime}{fieldName}[]{\gdef\animeBtnFieldName{#1}}
nFrames 2262 \define@key{btnAnime}{ocgAnimeName}[]{\gdef\animeBtnFieldName{#1}}
2263 \define@key{btnAnime}{iconName}[]{\gdef\animeBtnBaseName{#1}}
2264 \define@key{btnAnime}{speed}[200]{\gdef\animeBtnSpeed{#1}}
2265 \define@key{btnAnime}{nFrames}[]{\gdef\nFrames{#1}}

autorun Additional keys
autopause 2266 \define@boolkey{btnAnime}{autorun}[true]{}
autoplayevent 2267 \define@boolkey{btnAnime}{autoplayevent}[true]{}
autopauseevent 2268 \define@choicekey+{btnAnime}{autoplayevent}[\val\nr]%
poster 2269 {pageopen,pagevisible}[pageopen]{\ifcase\nr\relax
2270 \def\ae@bp@autoplayevent{\AAPageOpen}\or
2271 \def\ae@bp@autoplayevent{\AAPageVisible}\fi}{}%
2272 \def\ae@bp@autoplayevent{\AAPageOpen}
2273 \define@choicekey+{btnAnime}{autopauseevent}[\val\nr]%
2274 {pageclose,pageinvisible}[pageclose]{\ifcase\nr\relax
2275 \def\ae@bp@autopauseevent{\AAPageClose}\or
2276 \def\ae@bp@autopauseevent{\AAPageInvisible}\fi}{}%
2277 \def\ae@bp@autopauseevent{\AAPageClose}
2278 \define@choicekey+{btnAnime}{poster}[\val\nr]%
2279 {first,last,none}[first]{\edef\ae@bp@btnAnime@poster{\nr}}{}

```

(Needs FIX) Behavior of the poster. When `poster=first` this is OK, no bugs. When `poster=last` and the animation begins (at least for button) the animation does not appear to be running until the last frame, then it starts up. In this case, we display the last frame but when play is pressed the animation is ‘cleared’ and begins from frame 1.

`ctrlwidth` The following are key-values for the control buttons, in addition to `ctrlwidth`  
`ctrlheight` and `ctrlheight`, we also include `ctrlbdrywidth` and `ctrlbdrycolor` the latter  
`ctrlbdrywidth` two so we can get the spacing correct.

```

ctrlbdrycolor 2280 \define@key{btnAnime}{ctrlwidth}[18bp]{\def\btnAnimeCtrlW{#1}}
2281 \define@key{btnAnime}{ctrlheight}[9bp]{\def\btnAnimeCtrlH{#1}}
2282 \define@choicekey{btnAnime}{ctrlbdrywidth}[\val\nr]{none,%
2283     thin,medium,thick}[thin]{\edef\btnAnimeCtrlBdryWidth{\nr}}
(2017/03/08) Let ctrlbdrycolor accept xcolor, if loaded
2284 \define@key{btnAnime}{ctrlbdrycolor}[]{\def\ap@argi{#1}%
2285     \ifx\ap@argi\@empty\let\btnAnimeCtrlBdryColor\@empty\else
2286     \HyColor@IfXcolor{\HyColor@XZeroOneThreeFour{#1}
2287         {\btnAnimeCtrlBdryColor}{}}
2288     {\edef\btnAnimeCtrlBdryColor{#1}}\fi
2289 } % PDF color
2290 \let\animeSetupPresets\@empty

```

A command to set the space between control buttons, uses `\btnanimebtnsep`

```
2291 \newcommand{\@ctrlsbtnsep}{\cgbdry[\btnanimebtnsep]}
```

`\btnanimebtnsep` The space between buttons

`\vspacectrlsep` The vertical space between the bottom of the anime and the control button set.

`\btnanimerowsep` The vertical distance between two rows of control buttons.

```

2292 \newcommand{\btnanimebtnsep}{1bp}
2293 \newcommand{\vspacectrlsep}{3bp}
2294 \newcommand{\aep@vspacectrlsep}{\vspacectrlsep}
2295 \newcommand{\btnanimerowsep}{1bp}

```

### Definitions of “skins” may be redefined.

`\btnAnimeSkini` The definitions of the various control button layouts. These are normally not used  
`\btnAnimeSkinii` by the document author.

```

\btnAnimeSkiniii 2296 \newcommand{\btnAnimeSkini}{\ifKV@btnAnime@usetworows
\btnAnimeSkiniv 2297     \makebox[\linewidth][c]{%
\btnAnimeSkinv 2298     \btnAnimeGoToFirst\@ctrlsbtnsep\btnAnimeStepBack
\btnAnimeSkinvi 2299     \@ctrlsbtnsep\btnAnimePlayBack
2300     \@ctrlsbtnsep\btnAnimePause
2301     \@ctrlsbtnsep\btnAnimePlayForward
2302     \@ctrlsbtnsep\btnAnimeStepForward
2303     \@ctrlsbtnsep\btnAnimeGoToLast}\ifKV@btnAnime@nospeedcontrol\else
2304     \vcgbdry[\@btnanimerowsep]\makebox[\linewidth][c]{%
2305     \btnAnimePlus\@ctrlsbtnsep\btnAnimeMinus}\fi\else
2306     \makebox[\linewidth][s]{\hss
2307     \btnAnimeGoToFirst
2308     \@ctrlsbtnsep\btnAnimeStepBack
2309     \@ctrlsbtnsep\btnAnimePlayBack
2310     \@ctrlsbtnsep\btnAnimePause
2311     \@ctrlsbtnsep\btnAnimePlayForward

```

```

2312 \ctrlsbtnsep\btnAnimeStepForward
2313 \ctrlsbtnsep\btnAnimeGoToLast
2314 \ifKV@btnAnime@nospeedcontrol\else
2315 \kern\@spaceBtwnPMBtns\btnAnimePlus
2316 \ctrlsbtnsep\btnAnimeMinus\fi\hss}\fi}
2317 \newcommand{\btnAnimeSkiniii}{\ifKV@btnAnime@usetworows
2318 \makebox[\linewidth][c]{%
2319 \btnAnimeGoToFirst
2320 \ctrlsbtnsep\btnAnimePlayBack
2321 \ctrlsbtnsep\btnAnimePause
2322 \ctrlsbtnsep\btnAnimePlayForward
2323 \ctrlsbtnsep\btnAnimeGoToLast}%
2324 \ifKV@btnAnime@nospeedcontrol\else\vcbdry[\@btnanimerowsep]%
2325 \makebox[\linewidth][c]{\btnAnimePlus\ctrlsbtnsep
2326 \btnAnimeMinus}\fi\else\makebox[\linewidth][s]{\hss
2327 \btnAnimeGoToFirst\ctrlsbtnsep
2328 \btnAnimePlayBack\ctrlsbtnsep\btnAnimePause%
2329 \ctrlsbtnsep\btnAnimePlayForward\ctrlsbtnsep
2330 \btnAnimeGoToLast\ifKV@btnAnime@nospeedcontrol\else
2331 \kern\@spaceBtwnPMBtns\btnAnimePlus
2332 \ctrlsbtnsep\btnAnimeMinus\fi\hss}\fi}
2333 \newcommand{\btnAnimeSkiniii}{\ifKV@btnAnime@usetworows
2334 \makebox[\linewidth][c]{%
2335 \btnAnimeGoToFirst\ctrlsbtnsep\btnAnimePause
2336 \ctrlsbtnsep\btnAnimePlayForward}%
2337 \ifKV@btnAnime@nospeedcontrol\else
2338 \vcbdry[\@btnanimerowsep]\makebox[\linewidth][c]{%
2339 \btnAnimePlus\ctrlsbtnsep\btnAnimeMinus}\fi\else
2340 \makebox[\linewidth][s]{\hss
2341 \btnAnimeGoToFirst\ctrlsbtnsep\btnAnimePause
2342 \ctrlsbtnsep\btnAnimePlayForward
2343 \ifKV@btnAnime@nospeedcontrol\else\kern\@spaceBtwnPMBtns
2344 \btnAnimePlus\ctrlsbtnsep\btnAnimeMinus\fi\hss}\fi}
2345 \newcommand{\btnAnimeSkiniv}{%
2346 \makebox[\linewidth][c]{%
2347 \btnAnimeGoToFirst
2348 \ctrlsbtnsep\btnAnimeStepBack
2349 \ctrlsbtnsep\btnAnimeStepForward
2350 \ctrlsbtnsep\btnAnimeGoToLast}}
2351 \newcommand{\btnAnimeSkinv}{\PackageWarning{aeb_pro}
2352 {skin5 is undefined.\MessageBreak You may customize
2353 by defining skin5;\MessageBreak meanwhile, using default,
2354 skin1}\btnAnimeSkini}
2355 \newcommand{\btnAnimeSkinvi}{\PackageWarning{aeb_pro}
2356 {skin6 is undefined.\MessageBreak You may customize
2357 by defining skin6;\MessageBreak meanwhile, using default,
2358 skin1}\btnAnimeSkini}

```

**controls** The choice for the controls key. `skin5` and `skin6` default to `skin1`. These can be redefined for a custom configuration.

```

2359 \newcommand{\aepnumWidgetsFirstRow}[2]{\def\numWidgetsFirstRow{%
2360     \ifKV@btnAnime@nospeedcontrol#1\else
2361     \ifKV@btnAnime@usetworows#1\else#2\fi\fi}}
2362 \newcommand{\numWidgetsFirstRowV}{\aepnumWidgetsFirstRow{7}{9}}
2363 \newcommand{\numWidgetsFirstRowVI}{\aepnumWidgetsFirstRow{7}{9}}
2364 \define@choicekey+{btnAnime}{controls}[\val\nr]{%
2365     {none,skin1,skin2,skin3,skin4,skin5,skin6}[skin1]}{%
2366     \ifcase\nr\relax
2367         \let\aep@btn@anime@controls\@empty
2368         \def\widthFirstRow{Opt}\or
2369         \def\aep@btn@anime@controls{btnAnimeSkini}%
2370         \aepnumWidgetsFirstRow{7}{9}\or
2371         \def\aep@btn@anime@controls{btnAnimeSkinii}%
2372         \aepnumWidgetsFirstRow{5}{7}\or
2373         \def\aep@btn@anime@controls{btnAnimeSkiniii}%
2374         \aepnumWidgetsFirstRow{3}{5}\or
2375         \def\aep@btn@anime@controls{btnAnimeSkiniv}%
2376         \aepnumWidgetsFirstRow{4}{4}\or
2377         \def\aep@btn@anime@controls{btnAnimeSkinv}%
2378         \numWidgetsFirstRowV\or
2379         \def\aep@btn@anime@controls{btnAnimeSkinvi}%
2380         \numWidgetsFirstRowVI\fi
2381     }{}

```

usetworows The usetworows and nospeedcontrol key

```

nospeedcontrol 2382 \define@boolkey{btnAnime}{usetworows}[true]{}
2383 \define@boolkey{btnAnime}{nospeedcontrol}[true]{}

```

Set the defaults for button animation.

```

2384 \setkeys{btnAnime}{fieldName,iconName,speed,nFrames,type,poster,
2385     controls,usetworows=false,nospeedcontrol=false,ctrlwidth,ctrlheight,
2386     ctrlbdrywidth,ctrlbdrycolor}

```

```

2387 \newcommand{btnAnimePresets}[1]{\def\ae@btnAnimePresets{#1}}
2388 \def\ae@btnAnimePresets{}

```

`\ctrlButtonsWrapper` The `\ctrlButtonsWrapper` command takes one argument, and that is `\insert-CtrlButtons`. Redefine this command to, for example, place a `\colorbox` around the button set,

```

\renewcommand\ctrlButtonsWrapper}[1]{\colorbox{red}{#1}}

```

The default definition is...

```

2389 \newcommand{\ctrlButtonsWrapper}[1]{#1}

```

```

2390 % End animecommon segment

```

```

2391 \end{animecommon}

```

```

2392 \end{btnanime}

```

### 14.5.1 Define \btnAnime

`\btnAnime` The command `\btnAnime` supplies basic animation support using buttons. The command `\btnAnime` assumes the images are embedded in the document using `\embedMultiPageImages`, icon images for the controls are supplied by `\placeAnimeCtrlBtnFaces`.

```

2393 \newcommand{\btnAnime}[3]{\begingroup\animeSetup{#1}%
2394   \ifx\animeBtnFieldName\@empty\PackageError{aeb_pro}%
2395     {The fieldName key is required}\fi
2396   \ifx\animeBtnBaseName\@empty\PackageError{aeb_pro}%
2397     {The iconName key is required}\fi
2398   \ifx\nFrames\@empty\PackageError{aeb_pro}%
2399     {The nFrames key is required}\fi
2400   \parbox[t][#3]{#2}{%

Set the properties of the pushbutton to be created.
2401     \everyPushButton{\BC{} \BG{} \S{S} \autoCenter{n} \FB{true} \TP{1}%
2402     \Ff{\FfReadOnly} \presets{aeb@btnAnimePresets} \I{null}}%

Set the options for the first frame, it is either visible if poster=first or hidden,
otherwise.
2403     \edef\termLoop{\nFrames}\edef\firstBtnOpts{%
2404     \ifnum\aebp@btnAnime@poster=0 \else\noexpand\F{\FHHidden}\fi
2405     \noexpand\AA{\ifKV@btnAnime@autorun
2406     \aebp@autoplayevent{\noexpand\btnAnimePFAction}\fi
2407     \ifKV@btnAnime@autopause
2408     \aebp@autopauseevent{\noexpand\btnAnimePauseAction}\fi}}%

Begin with the first button.
2409     \makebox[Opt][1]{\expandafter\pushButton
2410     \expandafter[\firstBtnOpts]{\animeBtnFieldName.0}{#2}{#3}}%

If poster=last, decrement \termLoop by one.
2411     \ifnum\aebp@btnAnime@poster=1
2412     \count0=\nFrames\relax\advance\count0by-1
2413     \edef\termLoop{the\count0}\fi@tempcnta=1

Begin to ‘stack’ the buttons (first button already in place)
2414     \loop\ifnum\@tempcnta<\termLoop\relax
2415     \makebox[Opt][1]{\pushButton[\F{\FHHidden}]{%
2416     \animeBtnFieldName.the\@tempcnta}{#2}{#3}}%
2417     \advance\@tempcnta1\relax\repeat

Finished with the look, if poster=last, then we did not stack the last button, we
do it now.
2418     \ifnum\aebp@btnAnime@poster=1\relax\makebox[Opt][1]{%
2419     \pushButton{\animeBtnFieldName.the\termLoop}{#2}{#3}}%
2420     \fi\hfill\ifx\ae@btn@anime@controls\@empty\else
2421     \offinterlineskip\vcbdry[\vspacectrlsep]%

Now build the control buttons
2422     \everyPushButton{}\centering

```

```

2423     \ctrlButtonsWrapper{\insertCtrlButtons}\fi
2424   }%parbox
2425   \endgroup
2426 }

2427 % End btnanime segment
2428 </btnanime>
2429 <*ocganime>

```

### 14.5.2 Define ocgAnime

**ocgAnime** We upgrade the ocg animation to the same level of button animation. We define the `ocgAnime` environment. Between the `\begin{ocgAnime}` and `\end{ocgAnime}`, place the code for creating the animation. The only required parameters consist of the same key-value pairs accepted by `\btnAnime`. Some keys are not used, such as `iconName`. The fields `ocgAnimeName` and `nFrames` are required.

```

2430 \newenvironment{ocgAnime}[1]{\animeSetup{#1}%
2431   \ifx\animeBtnFieldName\@empty\PackageError{aeb_pro}%
2432     {The ocgAnimeName key is required}\fi
2433   \ifx\nFrames\@empty\PackageError{aeb_pro}%
2434     {The nFrames key is required}\fi
2435   \DeclareAnime{\animeBtnFieldName}{\animeBtnSpeed}{\nFrames}%
2436 }{%
2437   \edef\ae@ExpOpts{\noexpand\AA{\ifKV@btnAnime@autorun
2438     \ae@autoplayevent{\noexpand\btnAnimePFAction}\fi
2439     \ifKV@btnAnime@autopause\ae@autopauseevent{%
2440       \noexpand\btnAnimePauseAction}\fi}}\smash{\makebox[0pt][l]{%
2441     {\expandafter\pushButton\expandafter[\ae@ExpOpts\W\BC{}]\BG{}]\S{S}}%
2442     {\animeBtnFieldName-autorunpause}{0bp}{0bp}}}%
2443   \ifx\ae@btn@anime@controls\@empty\else\offinterlineskip
2444     \vcgBdry[\vspacectrlsep]\everyPushButton{%
2445       \centering\ctrlButtonsWrapper{\insertCtrlButtons}\fi
2446 }

2447 % End of ocganime segment
2448 </ocganime>
2449 <*animecommon>

```

**\btnAnimeCtrlPresets** Use `\btnAnimeCtrlPresets` to set the appearances for all these buttons below.

```

\btnAnimeGoToFirst The control buttons for button anime. These commands are not normally used
\btnAnimeStepBack by the document author, but can be used for a custom layout.
\btnAnimePlayBack 2450 \newcommand{\btnAnimeCtrlPresets}[1]{\def\ae@btnAnimeCtrlPresets{#1}}
\btnAnimePause 2451 \def\ae@btnAnimeCtrlPresets{}
\btnAnimePlayForward 2452 \newcommand{\btnAnimeGoToFirst}{%
\btnAnimeStepForward 2453   \pushButton[\TU{Go to first frame}\S{S}\BC{}
\btnAnimeGoToLast 2454     \FB{false}\I{null}\TP{1}\presets{\ae@btnAnimeCtrlPresets}
\btnAnimePlus 2455     \presets{\animeSetupPresets}\A{\JS{\btnAnimeFirstAction}}}%
\btnAnimeMinus 2456     {\animeBtnFieldName4BtnCtrl.animeFirst}%
2457     {\btnAnimeCtrlW}{\btnAnimeCtrlH}}%

```

```

2458 \newcommand{\btnAnimeStepBack}{%
2459     \pushButton[\TU{Step Back}\S{S}\BC{}
2460         \FB{false}\I{null}\TP{1}\presets{\aeb@btnAnimeCtrlPresets}
2461         \presets{\animeSetupPresets}\A{\JS{\btnAnimeSBAction}}}%
2462         {\animeBtnFieldName4BtnCtrl.animeSB}%
2463         {\btnAnimeCtrlW}{\btnAnimeCtrlH}}
2464 \newcommand{\btnAnimePlayBack}{%
2465     \pushButton[\TU{Play Backwards}\S{S}\BC{}
2466         \FB{false}\I{null}\TP{1}\presets{\aeb@btnAnimeCtrlPresets}
2467         \presets{\animeSetupPresets}\A{\JS{\btnAnimePBAction}}}%
2468         {\animeBtnFieldName4BtnCtrl.animePB}%
2469         {\btnAnimeCtrlW}{\btnAnimeCtrlH}}
2470 \newcommand{\btnAnimePause}{%
2471     \pushButton[\TU{Pause}\S{S}\BC{}
2472         \FB{false}\I{null}\TP{1}\presets{\aeb@btnAnimeCtrlPresets}
2473         \presets{\animeSetupPresets}\A{\JS{\btnAnimePauseAction}}}%
2474         {\animeBtnFieldName4BtnCtrl.animePause}%
2475         {\btnAnimeCtrlW}{\btnAnimeCtrlH}}
2476 \newcommand{\btnAnimePlayForward}{%
2477     \pushButton[\TU{Play Forwards}\S{S}\BC{}
2478         \FB{false}\I{null}\TP{1}\presets{\aeb@btnAnimeCtrlPresets}
2479         \presets{\animeSetupPresets}\A{\JS{\btnAnimePFAction}}}%
2480         {\animeBtnFieldName4BtnCtrl.animePF}%
2481         {\btnAnimeCtrlW}{\btnAnimeCtrlH}}
2482 \newcommand{\btnAnimeStepForward}{%
2483     \pushButton[\TU{Step Forward}\S{S}\BC{}
2484         \FB{false}\I{null}\TP{1}\presets{\aeb@btnAnimeCtrlPresets}
2485         \presets{\animeSetupPresets}\A{\JS{\btnAnimeSFAction}}}%
2486         {\animeBtnFieldName4BtnCtrl.animeSF}%
2487         {\btnAnimeCtrlW}{\btnAnimeCtrlH}}
2488 \newcommand{\btnAnimeGoToLast}{%
2489     \pushButton[\TU{Go to last frame}\S{S}\BC{}
2490         \FB{false}\I{null}\TP{1}\presets{\aeb@btnAnimeCtrlPresets}
2491         \presets{\animeSetupPresets}\A{\JS{\btnAnimeLastAction}}}%
2492         {\animeBtnFieldName4BtnCtrl.animeLast}%
2493         {\btnAnimeCtrlW}{\btnAnimeCtrlH}}
2494 % These last two do not take an icon.
2495 \newcommand{\btnAnimePlus}{%
2496     \pushButton[\TU{Increase speed\r Shift-click to restore
2497         default speed}%\textFont{HeBo}\CA{+}
2498         \S{S}\BC{}\FB{false}\I{null}\TP{1}
2499         \presets{\aeb@btnAnimeCtrlPresets}
2500         \presets{\animeSetupPresets}\A{\JS{\btnAnimePlusAction}}}%
2501         {\animeBtnFieldName4BtnCtrl.animePlus}%
2502         {\btnAnimeCtrlW}{\btnAnimeCtrlH}}
2503 \newcommand{\btnAnimeMinus}{%
2504     \pushButton[\TU{Decrease speed\r Shift-click to restore
2505         default speed}%\textFont{HeBo}\uCA{FEFF2212}
2506         \S{S}\BC{}\FB{false}\I{null}\TP{1}
2507         \presets{\aeb@btnAnimeCtrlPresets}

```



```

2508     \presets{\animeSetupPresets}\A{\JS{\btnAnimeMinusAction}}}%
2509     {\animeBtnFieldName4BtnCtrl.animeMinus}%
2510     {\btnAnimeCtrlW}{\btnAnimeCtrlH}}

```

The following are the field events for the controls for \btnAnime and acgAnime.

```

2511 \newcommand{\btnAnimeFirstAction}{%
2512   if (typeof \animeBtnFieldName\space == "object")\r\t
2513     btnAnimeFirst(\animeBtnFieldName);
2514 }
2515 \newcommand{\btnAnimeSBAction}{%
2516   if ( typeof \animeBtnFieldName\space=="undefined")\r\t
2517     var \animeBtnFieldName= new Object();\r
2518     btnAnimePlay(\animeBtnFieldName, "\animeBtnFieldName",%
2519       "\animeBtnBaseName", \animeBtnSpeed, \nFrames, false,%
2520       \aebp@btnAnime@type, true);
2521 }
2522 \newcommand{\btnAnimePBAction}{%
2523   if ( typeof \animeBtnFieldName\space=="undefined")\r\t
2524     var \animeBtnFieldName= new Object();\r
2525     btnAnimePlay(\animeBtnFieldName, "\animeBtnFieldName",%
2526       "\animeBtnBaseName", \animeBtnSpeed, \nFrames, false,%
2527       \aebp@btnAnime@type, false);
2528 }
2529 \newcommand{\btnAnimePauseAction}{%
2530   if ((typeof \animeBtnFieldName\space == "object")&&%
2531     \animeBtnFieldName.animeStart)\r\t
2532     btnAnimePause(\animeBtnFieldName);
2533 }
2534 \newcommand{\btnAnimePFAction}{%
2535   if ( typeof \animeBtnFieldName\space=="undefined")\r\t
2536     var \animeBtnFieldName= new Object();\r
2537     btnAnimePlay(\animeBtnFieldName, "\animeBtnFieldName",%
2538       "\animeBtnBaseName", \animeBtnSpeed, \nFrames, true,%
2539       \aebp@btnAnime@type, false);
2540 }
2541 \newcommand{\btnAnimeSFAction}{%
2542   if ( typeof \animeBtnFieldName\space=="undefined")\r\t
2543     var \animeBtnFieldName= new Object();\r
2544     btnAnimePlay(\animeBtnFieldName, "\animeBtnFieldName",%
2545       "\animeBtnBaseName", \animeBtnSpeed, \nFrames, true,%
2546       \aebp@btnAnime@type, true);
2547 }
2548 \newcommand{\btnAnimeLastAction}{%
2549   if (typeof \animeBtnFieldName\space == "object")\r\t
2550     btnAnimeLast(\animeBtnFieldName);
2551 }
2552 \newcommand{\btnAnimePlusAction}{%
2553   if (typeof \animeBtnFieldName\space == "object")\r\t
2554     btnAnimePlus(\animeBtnFieldName, \animeBtnSpeed);
2555 }

```

```

2556 \newcommand{\btnAnimeMinusAction}{%
2557     if (typeof \animeBtnFieldName\space == "object")\r\t
2558         btnAnimeMinus(\animeBtnFieldName,\animeBtnSpeed);
2559 }

```

The supporting JavaScript. This set of JavaScript functions support both button anime and ocg anime (the \btnAnime command and the ocgAnime environment).

```

2560 \begin{insDLJS}[_loadBtnAnimeCtrls]{btnanime}
2561     {AeB Pro: Button and Ocg Anime Controls}
2562 var _loadBtnAnimeCtrls=true;
2563 var activeAnimes=new Array();

```

This function may be called with a single argument oAnime when the anime is playing.

```

2564 function btnAnimePlay(oAnime,fieldName,baseFrameName,%
2565 nSpeed,nFrames,bForward,type,bStep) {
2566     // ocgAnime is characterized by baseFrameName=""
2567     if ( oAnime.animeStart && !oAnime.animePause ) return;
2568     var callingFieldName=event.target.name;
2569     var re=/\.anime(PB|PF)/;
2570     var doRebuild=( event.shift && re.test(callingFieldName) );
2571     if ( (arguments.length>1) && %
2572 ( (typeof activeAnimes[fieldName] == "undefined")|| doRebuild) ) {
2573         activeAnimes[fieldName]=oAnime;
2574         oAnime.isOcgAnime=(baseFrameName==""); // ocg change
2575         oAnime.fieldName=fieldName;
2576         oAnime.baseFrameName=baseFrameName;
2577         oAnime.numSecs=nSpeed;
2578         oAnime.nFrames=nFrames;
2579         oAnime.type=type;
2580         oAnime.bStep=bStep;
2581         oAnime.aMyIcons = new Array();
2582         oAnime.count = 0;
2583         if ( !oAnime.isOcgAnime ) { // ocg change
2584             for ( var i=0; i < nFrames; i++) {
2585                 oAnime.aMyIcons[i] = this.getField(fieldName+"."+i);

```

(2015/08/09) When starting for the first time, be sure initial frame showing is the first frame 0.

```

2586                 oAnime.aMyIcons[i].display=(i==0)?display.visible:%
2587 display.hidden;
2588             }
2589         }
2590     }
2591     if (arguments.length>1) {
2592         oAnime.bForward=bForward;
2593         oAnime.bStep=bStep;
2594     }
2595     if (oAnime.bStep) {

```

```

2596         this.delay=true
2597         if ( oAnime.isOcgAnime ) // ocg change
2598             getxBld("anime!!"+oAnime.fieldName+"\#"%)
2599 +(oAnime.count+1)).state=false; // ocg
2600         else
2601             oAnime.aMyIcons[oAnime.count].display=display.hidden;
2602         var currCount=oAnime.count;
2603         if (oAnime.bForward) {
2604             oAnime.count++;
2605             if (oAnime.count==oAnime.nFrames) // hit upper boundary
2606                 oAnime.count=(oAnime.type==0)?0:currCount;
2607         } else {
2608             oAnime.count--;
2609             if (oAnime.count<0)
2610                 oAnime.count=(oAnime.type==0)?(oAnime.nFrames-1):0;
2611         }
2612         if ( oAnime.isOcgAnime ) // ocg change
2613             getxBld("anime!!"+oAnime.fieldName+"\#"%)
2614 +(oAnime.count+1)).state=true; // ocg
2615         else
2616             oAnime.aMyIcons[oAnime.count].display=display.visible;
2617         this.delay=false;
2618     } else {
2619         var args="\ "+oAnime.fieldName+"\ ",oAnime.type;
2620         oAnime.run = app.setInterval(%
2621 "btnAnimeNewFrame("+args+")",oAnime.numSecs);
2622         oAnime.animeStart=true;
2623         oAnime.animePause=false;
2624     }
2625 }
2626 function btnAnimeFirst(oAnime)
2627 {
2628     try { app.clearInterval(oAnime.run);} catch(e){}
2629     oAnime.animeStart=false;
2630     oAnime.animePause=false;
2631     if ( oAnime.isOcgAnime )
2632         getxBld("anime!!"+oAnime.fieldName+"\#"%)
2633 +(oAnime.count+1)).state=false; // ocg
2634     else
2635         oAnime.aMyIcons[oAnime.count].display=display.hidden;
2636     oAnime.count=0;
2637     if ( oAnime.isOcgAnime )
2638         getxBld("anime!!"+oAnime.fieldName+"\#"%)
2639 +(oAnime.count+1)).state=true; // ocg
2640     else
2641         oAnime.aMyIcons[0].display=display.visible;
2642 }
2643 function btnAnimeLast(oAnime)
2644 {
2645     try { app.clearInterval(oAnime.run);} catch(e){}

```

```

2646     oAnime.animeStart=false;
2647     oAnime.animePause=false;
2648     var l = oAnime.nFrames-1;
2649     if ( oAnime.isOcgAnime )
2650         getxBld("anime!!"+oAnime.fieldName+"\#"%
2651 +(oAnime.count+1)).state=false; // ocg
2652     else
2653         oAnime.aMyIcons[oAnime.count].display=display.hidden;
2654     oAnime.count=l;
2655     if ( oAnime.isOcgAnime )
2656         getxBld("anime!!"+oAnime.fieldName+"\#"%
2657 +(l+1)).state=true; // ocg
2658     else
2659         oAnime.aMyIcons[l].display=display.visible;
2660 }
2661 function btnAnimePause(oAnime)
2662 {
2663     try { app.clearInterval(oAnime.run);} catch(e){}
2664     oAnime.animePause=true;
2665 }
2666 function btnAnimePlus(oAnime,nDefSpeed){
2667     var isPlaying=(oAnime.animeStart&&!oAnime.animePause);
2668     if (isPlaying) btnAnimePause(oAnime);
2669     if ( event.shift ) oAnime.numSecs=nDefSpeed;
2670     else oAnime.numSecs -= 50;
2671     if (oAnime.numSecs<10) oAnime.numSecs=10
2672     if (isPlaying) btnAnimePlay(oAnime);
2673 }
2674 function btnAnimeMinus(oAnime,nDefSpeed){
2675     var isPlaying=(oAnime.animeStart&&!oAnime.animePause);
2676     if (isPlaying) btnAnimePause(oAnime);
2677     if ( event.shift ) oAnime.numSecs=nDefSpeed;
2678     else oAnime.numSecs += 50;
2679     if (isPlaying) btnAnimePlay(oAnime);
2680 }
2681 function btnAnimeNewFrame(fieldName, type)
2682 {
    type=0,2,3: loop(0), palindrome(1), stopatboundary(2)
2683     var oAnime=activeAnimes[fieldName];
2684     this.delay=true
2685     if ( oAnime.isOcgAnime ) // ocg change
2686         getxBld("anime!!"+oAnime.fieldName+"\#"%
2687 +(oAnime.count+1)).state=false;
2688     else
2689         oAnime.aMyIcons[oAnime.count].display=display.hidden;
2690     var currCount=oAnime.count;
2691     var bBdaryMet=false;
2692     if (oAnime.bForward) {
2693         oAnime.count++;

```

```

2694         if (oAnime.count==oAnime.nFrames) {
2695             if(type==0) oAnime.count \%= oAnime.nFrames;
2696             else if(type==1) {
2697                 oAnime.count=currCount;
2698                 oAnime.count--;
2699                 oAnime.bForward=false;
2700             } else {
2701                 bBdaryMet=true;
2702                 oAnime.count=currCount;
2703             }
2704         }
2705     } else {
2706         oAnime.count--;
2707         if (oAnime.count<0) {
2708             if(type==0) oAnime.count=(oAnime.nFrames-1);
2709             else if(type==1) {
2710                 oAnime.count=currCount;
2711                 oAnime.count++;
2712                 oAnime.bForward=true;
2713             } else { // stop
2714                 bBdaryMet=true;
2715                 oAnime.count=currCount;
2716             }
2717         }
2718     }
2719     if ( oAnime.isOcgAnime) // ocg change
2720         getxBld("anime!!"+oAnime.fieldName+"\#"%
2721 +(oAnime.count+1)).state=true;
2722     else
2723         oAnime.aMyIcons[oAnime.count].display=display.visible;
2724     this.delay=false;
2725     if ( type==2 && bBdaryMet ) {
2726         try { app.clearInterval(oAnime.run);} catch(e){}
2727         // try { app.clearTimeout(timeout);} catch(e){}
2728         oAnime.animeStart=false;
2729         oAnime.animePause=false;
2730     }
2731 }
2732 \end{insDLJS}

2733 % End of animecommon segment
2734 \end{animecommon}

2735 % Begin package segment
2736 \begin{package}
2737 \ap@restoreCats
2738 \aeb@InputAeBPro
2739 \end{package}

```

## 15 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	
\!	142, 144
\#	544, 589, 590, 592, 606, 607, 610, 628, 637, 1871, 2598, 2613, 2632, 2638, 2650, 2656, 2686, 2720
\\$	1871
\%	1871, 1996, 2022, 2695
\:	1397, 1398
\@defineJS	1282, 1294
@defineJS (environment)	40
\@SOC	435
\@activechar@info	1345
\@aeb@InputUnicodes	86, 92
\@aeb@Close	1357, 1358, 1361, 1365, 1372
\@aeb@Open	1347, 1348, 1351, 1355, 1371
\@aeb@SimpleNesterror	437, 455
\@auxout	1700
\@bsphack	1927
\@btnanimerowsep	2239, 2241, 2304, 2324, 2338
\@ctrlsbtnsep	2291, 2298–2303, 2305, 2308– 2313, 2316, 2320–2323, 2325, 2327–2329, 2332, 2335, 2336, 2339, 2341, 2342, 2344, 2348–2350
\@defineJS	1260, 1267, 1276, 1287, 1454, 1459, 1463, 1467, 1471, 1475, 1479
\@dipArgs	2060, 2063
\@eqlinktcolor	1844, 1856
\@everyPageClosefalse	1257, 1343
\@everyPageClosetrue	1267, 1368
\@everyPageOpenfalse	1255, 1342
\@everyPageOpentrue	1260
\@fieldname	2092, 2093, 2128, 2129, 2204, 2205
\@fourthoffive	1388
\@goprofalse	58
\@goprotrue	57
\@handledblquotes	1865, 1866, 1870
\@ifl@t@r	224
\@initdfnJS	1321
\@isfirstslash	1711, 1714
\@let@token	1393, 1398, 1400
\@makeoother	144, 1871, 1881, 1886, 1928
\@empty	1776
\@mypage	1059, 1061
\@mytoks	1327, 1330, 1331, 1334
\@onlypreamble	519, 721, 946, 1686, 2105, 2109, 2161, 2197
\@pathsep	2183, 2185–2193
\@pdfm@mark	755, 794, 934, 1028, 1052, 1378, 1512, 1679
\@pdfstartpage	730
\@pdfstartview	729
\@rgi	203, 204, 210
\@rgii	203, 209, 211, 214, 216, 218, 221, 222, 224–228, 231
\@rgiii	207, 220
\@secondoffive	1386
\@setDocActions	1488, 1515
\@spaceBtwnPMBtns	2254, 2257, 2315, 2331, 2343
\@sptoken	1393
\@tempExp	2230
\@tempexp	2067, 2068
\@tforExp	2060, 2061
\@thisattachnum	1892, 1903, 1911, 1913, 1921, 1922
\@tmpExp	2233, 2235, 2236, 2238
\@useacrobatfalse	55
\@useacrobattrue	56, 58, 149, 152
\@vrbsetupdfnJS	1323
\{	1522
\}	1627
\^	1929
\_	13
<b>A</b>	
\A	549, 558, 567, 1848, 1860, 2455, 2461, 2467, 2473, 2479, 2485, 2491, 2500, 2508
\AA	427, 2405, 2437
\AAMouseEnter	427
\AAMouseExit	428
\AAPageClose	2275, 2277
\AAPageInvisible	2276
\AAPageOpen	2270, 2272
\AAPageVisible	2271
\active	1864, 1865, 1929
\additionalOpenAction	<u>722</u>
\addJST@PageClose	1287, 1290, 1299, 1310, 1312, 1314
\addJST@PageOpen	1276, 1278, 1298, 1309, 1311, 1313
\addJSTexHelpEnter	176, 413
\addJSTexHelpExit	177, 414
addJSToPageClose (environment)	<u>1255</u>
addJSToPageCloseAt (environment)	<u>1255</u>
addJSToPageOpen (environment)	<u>1255</u>
addJSToPageOpenAt (environment)	<u>1255</u>
\addSpaceBtwnPMBtns	2245, 2256

<code>\addtocounter</code> .....	1702, 1703	<code>\aeb@InputAeBPro</code> .....	60, 2738
<code>\addtoDocOpen</code> .....	722, 729, 860	<code>\aeb@inputAttachments</code> .....	1633
<code>\addtoOptAttachments</code> .....	1524, 1542, 1544	<code>\aeb@InputUnicodes</code> .....	90, 92, 100, 303
<code>\AddToShipoutPicture</code> .....	1318	<code>\aeb@isFromAutoLabelNum</code> ...	1876, 1880, 1883, 1909
<code>\addWatermarkFromFile</code> .....	1972	<code>\aeb@IWVO</code> .....	309,
<code>\addWatermarkFromText</code> .....	1973	1521, 1522, 1585, 1589, 1610, 1614, 1628, 1630	
<code>\aeb@@@@parse</code> .....	1413, 1422	<code>\aeb@labelName</code> .....	1873, 1908
<code>\aeb@@@parse</code> .....	1412, 1414	<code>\aeb@labelNamestar</code> .....	1875, 1919
<code>\aeb@@@parse</code> .....	1410, 1411	<code>\aeb@layoutMagMag</code> .	687–691, 693, 694, 889, 891,
<code>\aeb@@@parse</code> .....	1409, 1410	894, 897, 902, 905, 908, 913, 918, 920, 923, 927	
<code>\aeb@addActionCode</code> .....	723, 735, 862	<code>\aeb@layoutMagNaviTab</code> .	671, 675, 876, 877, 884, 885
<code>\aeb@addOpenAction</code> .....	722, 746, 753,	<code>\aeb@layoutMagPage</code> .....	696, 697,
760, 892, 896, 898, 906, 911, 914, 921, 925, 928		699, 888, 895, 897, 901, 910, 913, 917, 924, 927	
<code>\aeb@addOpenActionArg</code> .....	724, 733, 742, 750, 757	<code>\aeb@layoutMagPageLayout</code> .....	678, 682, 881, 882
<code>\aeb@ahref@setlink</code> .....	1828, 1840, 1842	<code>\aeb@lowerlimit</code> .	1422, 1429, 1430, 1432–1434, 1444
<code>\aeb@arg</code> .....	965, 967, 975, 980, 988, 1087, 1099	<code>\aeb@next</code> ..	1412, 1413, 1894, 1897, 1900, 1904–1906
<code>\aeb@argi</code> .....	202, 648, 660	<code>\aeb@NoTransition</code> .....	947, 967
<code>\aeb@argii</code> .....	202, 649, 650, 653, 662, 663, 666	<code>\aeb@ocOntrue</code> .....	349
<code>\aeb@attachments</code> .....	76, 77,	<code>\aeb@openatpageChoice</code> .....	695, 698
158, 160, 1546, 1547, 1564, 1598, 1599, 1601, 1689		<code>\aeb@pack@def</code> .....	29, 32
<code>\aeb@attachments@xref</code> .....	1938–1940, 1947	<code>\aeb@Page@Dur</code> .....	1018, 1042, 1100, 1101
<code>\aeb@attachments@write</code> .....	1518–1520, 1631	<code>\aeb@PageClosefalse</code> .....	1346
<code>\aeb@attachsource</code> .....		<code>\aeb@PageClosetrue</code> .....	1360, 1364
..... 74, 75, 157, 159, 1563, 1579, 1580, 1688		<code>\aeb@PageOpenfalse</code> .....	1346
<code>\aeb@autoLabelNum</code> .....	1890, 1891	<code>\aeb@PageOpentru</code> .....	1350, 1354, 1367
<code>\aeb@autoLabelNumstar</code> .....	1890, 1902	<code>\aeb@pagetranschoices</code> .....	
<code>\aeb@border@parameters</code> ....	1807, 1810, 1846, 1858	. 1018, 1023, 1026, 1029, 1042, 1047, 1050, 1053	
<code>\aeb@btnAnimeCtrlPresets</code> ....	2450, 2451, 2454,	<code>\aeb@parse</code> .....	1406, 1407, 1421, 1451
2460, 2466, 2472, 2478, 2484, 2490, 2499, 2507		<code>\aeb@parsecnt</code> .....	
<code>\aeb@btnAnimePresets</code> .....	2387, 2388, 2402	.. 1328, 1329, 1337, 1390, 1417, 1418, 1420,	
<code>\aeb@CatEntry</code> .....	863, 932–934	1425, 1426, 1429–1433, 1442–1445, 1447, 1449,	
<code>\aeb@childof</code> .....	96–98	1450, 1600, 1606–1609, 1612, 1617, 1618, 1621	
<code>\aeb@childofInput</code> .....	95, 97, 101	<code>\aeb@parsetoks</code> .....	1059, 1327,
<code>\aeb@driver</code> .....	40, 54, 148, 151	1389, 1404, 1406, 1408, 1418, 1419, 1447, 1448	
<code>\aeb@exiii</code> .....	1383, 1384, 1386, 1388	<code>\aeb@pcks</code> .....	24, 27, 28, 30, 31
<code>\aeb@fsBGColor</code> .....	976–978, 1226	<code>\aeb@pdfpack@initview</code> .	1646, 1647, 1657, 1664, 1668
<code>\aeb@fsclickAdv</code> .....	998, 999, 1228	<code>\aeb@pdfpack@viewmode</code> .....	1641–1644, 1672
<code>\aeb@fscursor</code> .....	989, 990, 993, 1229	<code>\aeb@pdfpackage@initview</code> .....	1665, 1667, 1672
<code>\aeb@fsEscape</code> .....	995, 996, 1233	<code>\aeb@PDFPackageCode</code> .....	1670, 1675, 1677, 1679
<code>\aeb@fsFS</code> .....	1003, 1234	<code>\AEB@psMrk</code> .....	725, 737,
<code>\aeb@fsLoop</code> .....	985, 986, 1227	739, 782, 784, 931, 1022, 1046, 1374, 1508, 1674	
<code>\aeb@fstimeDelay</code> .....	981–983, 1230	<code>\aeb@removeLastcomma</code> .....	1404, 1408
<code>\aeb@fsTran</code> .....	966, 968, 970, 973, 1225	<code>\aeb@requiredVersionMsg</code> .....	815, 843
<code>\aeb@fsusePageTiming</code> .....	1008, 1009, 1232	<code>\aeb@requiredVersionMsgRedirect</code> .....	817, 852
<code>\aeb@fsuseTimer</code> .....	1005, 1006, 1231	<code>\aeb@reqVerMsgC</code> .....	816, 846
<code>\aeb@image@cnt</code> .....	2045, 2069–2071	<code>\aeb@reqVerMsgR</code> .....	818, 847
<code>\aeb@initahref</code> .....	1792, 1821, 1833, 1854	<code>\aeb@reqVerMsgW</code> .....	816, 818, 847
<code>\aeb@initAttachWrite</code> .....	1517, 1565, 1568	<code>\aeb@setahyperrefDefaults</code> .....	
<code>\aeb@Input@xref</code> .....	1948, 1949	..... 1788, 1789, 1791, 1796, 1798	
<code>\aeb@InputAeBLayers</code> .....	66, 69, 170	<code>\aeb@SetCatalog</code> .....	859, 937

\aeb@setPDFPackage	1674, 1676, 1678, 1680	\aeb@Trans@SplitHorizontalOut	1104
\aeb@SimpleOCOnfalse	434	\aeb@Trans@SplitVerticalIn	1105
\aeb@SimpleOCOntrue	435	\aeb@Trans@SplitVerticalOut	1106
\aeb@TaggedPDF	719, 720, 879	\aeb@Trans@UncoverDown	1144
\aeb@testdriver	147, 148, 150, 151	\aeb@Trans@UncoverLeft	1142
\aeb@TranChoice	1019, 1032, 1043, 1056, 1084, 1086	\aeb@Trans@UncoverLeftDown	1146
\aeb@Trans@Blend	1152	\aeb@Trans@UncoverLeftUp	1147
\aeb@Trans@BlindsHorizontal	1107	\aeb@Trans@UncoverRight	1143
\aeb@Trans@BlindsVertical	1108	\aeb@Trans@UncoverRightDown	1148
\aeb@Trans@BoxIn	1109	\aeb@Trans@UncoverRightUp	1149
\aeb@Trans@BoxOut	1110	\aeb@Trans@UncoverUp	1145
\aeb@Trans@CombHorizontal	1201	\aeb@Trans@WipeDown	1113
\aeb@Trans@CombVertical	1203	\aeb@Trans@WipeLeft	1111
\aeb@Trans@CoverDown	1136	\aeb@Trans@WipeLeftDown	1161
\aeb@Trans@CoverLeft	1134	\aeb@Trans@WipeLeftUp	1163
\aeb@Trans@CoverLeftDown	1138	\aeb@Trans@WipeRight	1112
\aeb@Trans@CoverLeftUp	1139	\aeb@Trans@WipeRightDown	1165
\aeb@Trans@CoverRight	1135	\aeb@Trans@WipeRightUp	1167
\aeb@Trans@CoverRightDown	1140	\aeb@Trans@WipeUp	1114
\aeb@Trans@CoverRightUp	1141	\aeb@Trans@ZoomInDown	1169
\aeb@Trans@CoverUp	1137	\aeb@Trans@ZoomInLeft	1171
\aeb@Trans@Dissolve	1115	\aeb@Trans@ZoomInLeftDown	1173
\aeb@Trans@Dur	1019, 1043, 1088, 1089	\aeb@Trans@ZoomInLeftUp	1175
\aeb@Trans@Fade	1150	\aeb@Trans@ZoomInRight	1177
\aeb@Trans@FlyIn	1128	\aeb@Trans@ZoomInRightDown	1179
\aeb@Trans@FlyInDown	1122	\aeb@Trans@ZoomInRightUp	1181
\aeb@Trans@FlyInLeft	1121	\aeb@Trans@ZoomInUp	1183
\aeb@Trans@FlyInRight	1120	\aeb@Trans@ZoomOutDown	1185
\aeb@Trans@FlyInUp	1123	\aeb@Trans@ZoomOutLeft	1187
\aeb@Trans@FlyOut	1129	\aeb@Trans@ZoomOutLeftDown	1189
\aeb@Trans@FlyOutDown	1126	\aeb@Trans@ZoomOutLeftUp	1191
\aeb@Trans@FlyOutLeft	1125	\aeb@Trans@ZoomOutRight	1193
\aeb@Trans@FlyOutRight	1124	\aeb@Trans@ZoomOutRightDown	1195
\aeb@Trans@FlyOutUp	1127	\aeb@Trans@ZoomOutRightUp	1197
\aeb@Trans@GlitterDown	1117	\aeb@Trans@ZoomOutUp	1199
\aeb@Trans@GlitterRight	1116	\aeb@uiopts	656, 658
\aeb@Trans@GlitterRightDown	1118	\aeb@uioptsHideMenubar	872, 873
\aeb@Trans@NoTransition	1102	\aeb@uioptsHideToolbar	870, 871
\aeb@Trans@PushDown	1132	\aeb@uioptsHideWindowUI	874, 875
\aeb@Trans@PushLeft	1130	\aeb@upperlimit	1422, 1424, 1431, 1435, 1443, 1450
\aeb@Trans@PushLeftDown	1153	\aeb@ViewerPassedRequirementJS	837, 857, 858
\aeb@Trans@PushLeftUp	1155	\aeb@windowopts	643, 646
\aeb@Trans@PushRight	1131	\aeb@windowoptsCenterWindow	866, 867
\aeb@Trans@PushRightDown	1157	\aeb@windowoptsDisplayDocTitle	868, 869
\aeb@Trans@PushRightUp	1159	\aeb@windowoptsFitWindow	864, 865
\aeb@Trans@PushUp	1133	\aeb@windowoptsFullScreen	883, 1001
\aeb@Trans@Random	1151	\aeb_mlink (option)	3
\aeb@Trans@Replace	1119	\aebAlternateDocumentURL	826, 828, 841
\aeb@Trans@Speed	1093–1095	\aebDeclareInitView@layoutmag	703, 704, 714
\aeb@Trans@SplitHorizontalIn	1103	\aebDeclareInitView@uioptions	709, 710, 716



<code>\aebDeclareInitView@windowoptions</code>	.. 706, 707, 715	<code>\aebpxBld@print</code>	..... 442, 451, 464–468, 471
<code>\aebp@addJStexHelpEnter</code>	..... 413, 416, 427	<code>\aebpxBld@visible</code>	..... 441, 472, 473, 477–480
<code>\aebp@addJStexHelpExit</code>	..... 414, 417, 429	<code>\aebsavehelp</code>	..... 310, 323, 325, 425, 430, 431
<code>\AEBP@aebpxBld@printfalse</code>	..... 460, 465, 468	<code>\aebwritelastpage</code>	..... 1699, 1702
<code>\AEBP@aebpxBld@printtrue</code>	..... 464, 467	<code>aebxmp (option)</code>	..... 3
<code>\aebp@app@title</code>	..... 807, 823, 831	<code>\aep@btn@anime@controls</code>	..... 2250, 2367, 2369, 2371, 2373, 2375, 2377, 2379, 2420, 2443
<code>\aebp@argi</code>	..... 808, 810	<code>\aep@ExpOpts</code>	..... 2437, 2441
<code>\aebp@attachments</code>	..... 1687–1690	<code>\aep@I</code>	..... 2169–2179, 2181
<code>\aebp@autopauseevent</code>	..... 2275–2277, 2408, 2439	<code>\aep@setAFirst</code>	..... 2169, 2185
<code>\aebp@autoplayevent</code>	..... 2270–2272, 2406, 2438	<code>\aep@setALast</code>	..... 2175, 2191
<code>\aebp@btnAnime@poster</code>	..... 2279, 2404, 2411, 2418	<code>\aep@setAMinus</code>	..... 2177, 2193
<code>\aebp@btnAnime@type</code>	... 2260, 2520, 2527, 2539, 2546	<code>\aep@setAPaus</code>	..... 2172, 2188
<code>\aebp@description</code>	1872, 1874, 1911, 1916, 1921, 1923	<code>\aep@setAPB</code>	..... 2171, 2187
<code>\aebp@dljstmp</code>	..... 764, 765	<code>\aep@setAPF</code>	..... 2173, 2189
<code>\aebp@dljstmpP</code>	..... 801, 802	<code>\aep@setAPlus</code>	..... 2176, 2192
<code>\aebp@DocActionsCode</code>	..... 1495, 1509, 1510, 1512	<code>\aep@setASB</code>	..... 2170, 2186
<code>\aebp@getFace</code>	..... 2053, 2093, 2129, 2205	<code>\aep@setASF</code>	..... 2174, 2190
<code>\aebp@getPlacement</code>	..... 2054, 2055	<code>\aep@thisAnime</code>	..... 2179, 2180
<code>\aebp@inputCommonAnimeCode</code>	..... 109, 115, 125	<code>\aep@vspacectrlsep</code>	..... 2294
<code>\aebp@isattach</code>	..... 156–158, 163	<code>\aepnumWidgetsFirstRow</code>	..... ..... 2359, 2362, 2363, 2370, 2372, 2374, 2376
<code>\aebp@labelname</code>	..... ..... 1882, 1884, 1887, 1913, 1916, 1918, 1922–1924	<code>\aftergroup</code>	..... 1264, 1271, 1279, 1291, 1948
<code>\aebp@NO</code>	19, 35, 36, 156, 727, 732, 773, 777, 860, 1405	<code>\afterRequirementPassedJS</code>	..... <u>722</u>
<code>\aebp@One</code>	..... 18, 1350, 1353, 1360, 1363, 1367–1369, 1688–1690, 1883, 1909	<code>\ahref@invisible@border</code>	..... 1731, 1733, 1812
<code>\aebp@PageActionCode</code>	..... 1370, 1375, 1377, 1379	<code>\ahref@border</code>	..... 1730, 1732, 1810
<code>\aebp@PageOpenCloseHandler</code>	..... <u>1300</u>	<code>\ahref@bordercolor</code>	1744, 1746, 1747, 1749, 1815, 1816
<code>\aebp@PageTransCnt</code>	..... ..... 1010, 1011, 1013, 1020, 1034, 1044, 1062	<code>\ahref@BS@Dict</code>	..... 1813, 1847, 1859
<code>\aebp@PageTransHandler</code>	..... 1316, 1318	<code>\ahref@dasharray</code>	..... 1805, 1808, 1814
<code>\aebp@passDriver</code>	..... 34–36, 45	<code>\ahref@dashedlinetype</code>	..... 1757, 1806
<code>\aebp@PassToHYIns</code>	..... 43, 46	<code>\ahref@dest</code>	..... 1717, 1718, 1822, 1826, 1834, 1838
<code>\aebp@PkgWarningI</code>	..... 192, 262, 271, 278, 284	<code>\ahref@dest@keyvalue</code>	.. 1823, 1826, 1835, 1838, 1848
<code>\aebp@PkgWarningII</code>	..... 195, 276, 282, 288	<code>\ahref@highlight</code>	..... 1738–1741, 1816
<code>\aebp@PkgWarningIII</code>	..... 198, 269	<code>\ahref@launch@type</code>	..... 1773, 1774, 1861
<code>\aebp@rightDelimiters</code>	.. 728, 743, 751, 758, 764, 765	<code>\ahref@linestyle@keyvalue</code>	.. 1753–1755, 1806, 1814
<code>\aebp@rightDelimitersP</code>	775, 785, 790, 795, 801, 802	<code>\ahref@linewidth</code>	..... 1761–1763, 1807, 1812, 1814
<code>\aebp@rse</code>	..... 1057, 1325, 1405	<code>\ahref@open@keyvalue</code>	..... 1722–1724, 1726, 1849
<code>\aebp@setCCOptions</code>	..... 22, 31	<code>\ahref@page</code>	..... 1708, 1709, 1824, 1836
<code>\aebp@Step@PageTransCnt</code>	..... 1011, 1016, 1040	<code>\ahref@preset</code>	..... 1775, 1776, 1802, 1804
<code>\aebp@uselayers</code>	..... 64, 66, 67, 106, 107, 169, 305	<code>\ahref@view@keyvalue</code>	.. 1712–1714, 1716, 1824, 1836
<code>\aebp@warnonly</code>	..... 806, 810	<code>\ahrefcolor</code>	... 55, 1777–1779, 1843, 1845, 1855, 1857
<code>\aebp@YES</code>	..... 19, 34, 45, 157, 158, 163, 253, 258, 290, 762, 799, 861, 1283, 1295, 1420, 1445	<code>\ahref@exafter</code>	..... 1384
<code>\aebp@Zero</code>	..... 18, 1345, 1687, 1876, 1880	<code>\ahypercolor</code>	... 1844, 1845, 1849, 1856, 1857, 1861
<code>\aebPageAction</code>	..... 1311, <u>1344</u>	<code>\ahyperextract</code>	..... <u>1852</u>
<code>\aebp@FA</code>	..... 727, 732, 762, 860	<code>\ahyperlink</code>	..... <u>1792</u>
<code>\aebp@FAP</code>	..... 773, 777, 799, 861	<code>\ahyperref</code>	..... <u>1792</u>
<code>\aebp@poptoks</code>	..... 726, 734, 740, 748, 755, 766	<code>\ahypertxtcolor</code>	..... 1777
<code>\aebp@poptoksP</code>	..... 774, 779, 784, 789, 794, 803	<code>\alnargi</code>	..... 1891, 1893, 1903, 1904
		<code>\alternateDocumentURL</code>	..... <u>722</u>
		<code>\animBaseName</code>	.. 542, 544, 552, 553, 561, 562, 570, 571



`\btnAnimeLastAction` ..... 2491, 2548  
`\btnAnimeMinus` 2305, 2316, 2326, 2332, 2339, 2344, 2450  
`\btnAnimeMinusAction` ..... 2508, 2556  
`\btnAnimePause` 2300, 2310, 2321, 2328, 2335, 2341, 2450  
`\btnAnimePauseAction` ..... 2408, 2440, 2473, 2529  
`\btnAnimePBAction` ..... 2467, 2522  
`\btnAnimePFAction` ..... 2406, 2438, 2479, 2534  
`\btnAnimePlayBack` .... 2299, 2309, 2320, 2328, 2450  
`\btnAnimePlayForward` .....  
..... 2301, 2311, 2322, 2329, 2336, 2342, 2450  
`\btnAnimePlus` 2305, 2315, 2325, 2331, 2339, 2344, 2450  
`\btnAnimePlusAction` ..... 2500, 2552  
`\btnAnimePresets` ..... 2387  
`\btnanimerowsep` ..... 2239, 2240, 2292  
`\btnAnimeSBAction` ..... 2461, 2515  
`\btnAnimeSFAction` ..... 2485, 2541  
`\btnAnimeSkini` ..... 2296, 2369  
`\btnAnimeSkinii` ..... 2296, 2371  
`\btnAnimeSkiniii` ..... 2296, 2373  
`\btnAnimeSkiniv` ..... 2296, 2375  
`\btnAnimeSkinv` ..... 2296, 2377  
`\btnAnimeSkinvi` ..... 2296, 2379  
`\btnAnimeStepBack` ..... 2298, 2308, 2348, 2450  
`\btnAnimeStepForward` ..... 2302, 2312, 2349, 2450

## C

`\CA` ..... 557, 2497  
`\canceleveryPageClose` ..... 1257  
`\canceleveryPageOpen` ..... 1255  
`center (key)` ..... 22  
`\centering` ..... 2250, 2422, 2445  
`\certifyInvisibleSign` ..... 2022  
`\cgbdry` ..... 2291  
`\chgDocObjectTo` ..... 1957  
`\clearAnimeBtn` ..... 189, 546  
`\clearpage` ..... 1702  
`clickAdv (key)` ..... 33  
`\color` ..... 315, 411  
`controls (key)` ..... 76  
`\createTemplate` ..... 1986  
`\csarg` ..... 483, 486–488,  
492, 493, 653, 666, 1020, 1033, 1044, 1061,  
1331, 1910, 1912, 1915, 1917, 1920, 1922–1924  
`ctrlbdrycolor (key)` ..... 74  
`ctrlbdrywidth (key)` ..... 74  
`\ctrlButtonsWrapper` ..... 2389, 2423, 2445  
`ctrlheight (key)` ..... 74  
`ctrlwidth (key)` ..... 74  
`\CurrentOption` ..... 6  
`cursor (key)` ..... 32

`\cxkvsetkeys` ..... 655, 668, 701

## D

`\DeclareAnime` ..... 187, 535, 2435  
`\declareImageAndPlacement` ..... 2044  
`\DeclareInitView` ..... 702  
`\DeclareJSHelper` 1963, 1972, 1973, 1977, 1981–1991  
`\declareMultiImages` ..... 2059, 2184  
`\DeclareOptionX` ..... 6,  
56, 58, 65, 73, 74, 76, 80, 83, 91, 97, 105, 113  
`\define@boolkey` ..... 984, 994, 997,  
1000, 1004, 1007, 2050, 2266, 2267, 2382, 2383  
`\define@choicekey` .....  
..... 37, 461, 474, 669, 676, 683, 948, 987,  
1067, 1090, 1638, 1719, 1727, 1735, 1750, 1758,  
1765, 1772, 2258, 2268, 2273, 2278, 2282, 2364  
`\definecolor` ..... 311  
`\defineJSArg` ..... 1322, 1325  
`\definePR` ..... 182, 485, 495  
`\defineRC` ..... 180, 483  
`\defUniStr` ..... 1879  
`dest (key)` ..... 54  
`\dev@addActionCode` ..... 772, 780, 861  
`\dev@addOpenAction` .....  
. 741, 743, 749, 751, 756, 758, 770, 787, 792, 797  
`\dev@addOpenActionArg` ..... 771, 778  
`\dev@addToDocOpen` ..... 770, 809, 1691  
`\developer@did@Print` ..... 1487, 1494, 1505  
`\developer@did@Save` ..... 1485, 1492, 1501  
`\developer@will@Close` ..... 1483, 1490, 1497  
`\developer@will@Print` ..... 1486, 1493, 1503  
`\developer@will@Save` ..... 1484, 1491, 1499  
`\dfnJSCR` ..... 1996, 2022  
`\did@Print` ..... 1479, 1481, 1494, 1505  
`\did@Save` ..... 1471, 1473, 1492, 1501  
`didPrint (environment)` ..... 1452  
`didSave (environment)` ..... 1452  
`\ding` ..... 314, 410  
`\dl@actualsize` ..... 687  
`\dl@fitheight` ..... 690  
`\dl@fitpage` ..... 688  
`\dl@fitvisible` ..... 691  
`\dl@fitwidth` ..... 689  
`\dl@mag` ..... 693  
`\dl@preProcDefns` ..... 735, 780, 1819, 1831, 1852  
`\dlfortypesetfalse` ..... 1322  
`dljslib (option)` ..... 3  
`\DLspecialDefs` ..... 1866, 1867  
`docassembly (environment)` ..... 1954  
`\docSaveAs` ..... 1983

\dospecials ..... 1928  
 \DQ ..... 1864, 1865  
 driver (option) ..... 4

## E

\e@OC ..... 439, 454  
 \earlyAttachForPkgs ..... 1648  
 \eBld ..... 179, 454  
 \edefaeb@parsetoks ..... 1418, 1419, 1447, 1448  
 eforms (option) ..... 3  
 \egroup ..... 761, 798, 1918, 1924  
 \embedMultiPageImages ..... 2110  
 embedonly (key) ..... 67  
 \end@defineJS ..... 1282, 1294  
 \end@defineJS ..... 1261, 1268, 1276,  
 1287, 1455, 1460, 1464, 1468, 1472, 1476, 1480  
 \endexecJS ..... 1956  
 \endverbatimwrite ..... 1934, 1945  
 environments:  
 @@defineJS ..... 40  
 addJSToPageClose ..... 1255  
 addJSToPageCloseAt ..... 1255  
 addJSToPageOpen ..... 1255  
 addJSToPageOpenAt ..... 1255  
 attachmentNames ..... 1936  
 didPrint ..... 1452  
 didSave ..... 1452  
 docassembly ..... 1954  
 ocgAnime ..... 2430  
 printRollover ..... 495  
 rollover ..... 495  
 willClose ..... 1452  
 willPrint ..... 1452  
 willSave ..... 1452  
 eq2db (option) ..... 3  
 \eq@pkgReq ..... 21, 208, 216  
 \eq@proofingfalse ..... 496, 498  
 \eq@proofingtrue ..... 497  
 \eq@theseOpts ..... 215, 218  
 \eqforpaperfalse ..... 495  
 escape (key) ..... 33  
 \every@PageClose ..... 1258, 1267, 1270, 1274, 1361, 1365  
 \every@PageOpen ..... 1256, 1260, 1263, 1273, 1351, 1355  
 \everyPushButton ..... 2401, 2422, 2444  
 \excludecomment ..... 306, 307, 505, 513  
 \execJS ..... 1955  
 \execjs ..... 253, 258, 290  
 \ExecuteOptionsX ..... 106, 129, 130, 132–136  
 \ExecuteOptionsXSAVE ..... 129, 132  
 \executeSave ..... 1989

exerquiz (option) ..... 3  
 \expand@setkeys ..... 1793, 1795, 1797, 1799, 1803, 1804  
 \extractPages ..... 1990

## F

\F ..... 2404, 2415  
 \factory@ahyperref@defaults ..... 1785, 1794  
 false (key) ..... 16  
 \FB ..... 2223, 2401, 2454,  
 2460, 2466, 2472, 2478, 2484, 2490, 2498, 2506  
 \Ff ..... 2223, 2402  
 \FfReadOnly ..... 2223, 2402  
 \FHidden ..... 2404, 2415  
 fieldName (key) ..... 74  
 \firstBtnOpts ..... 2403, 2410  
 firstpage (key) ..... 67  
 \firsttok ..... 1330, 1335  
 fit (key) ..... 22  
 \footnotesize ..... 314, 410  
 \forwardAnimeBtn ..... 190, 546  
 \frstPageSpec ..... 1283, 1295, 1405, 1420, 1445  
 \fs@will@Close .. 1454, 1456, 1463, 1465, 1490, 1497  
 fullscreen (key) ..... 22, 33  
 \futurelet ..... 1398, 1400

## G

\g@addto@macro . 734, 766, 779, 803, 1527, 2233, 2236  
 \getargsii ..... 202, 646, 658  
 \getcNameFromFile ..... 1528  
 \getdscripCont ..... 1872, 1885  
 \getdscripStrCont ..... 1874, 1888  
 gopro (option) ..... 6  
 goto (key) ..... 55  
 graphicxsp (option) ..... 3

## H

\H ..... 324, 426  
 \HandleDblQuotesfalse ..... 1869  
 \HandleDblQuotestrue ..... 1942  
 \headerps@out ..... 352  
 hidemenubar (key) ..... 22  
 hidetoolbar (key) ..... 22  
 hidewindowui (key) ..... 23  
 highlight (key) ..... 54  
 \Hy@pdfstringfalse ..... 336, 341, 526, 530  
 \Hy@pdfstringtrue ..... 337, 342  
 \HyColor@IfXcolor ..... 1745, 2286  
 \HyColor@XZeroOneThreeFour ..... 1745, 2286  
 \hyper@normalise ..... 1555, 1556, 1558  
 hyperref (option) ..... 3

## I

`\I` ..... 2224, 2402, 2454,  
2460, 2466, 2472, 2478, 2484, 2490, 2498, 2506  
`iconName` (key) ..... 74  
`\IconNameI` ..... 2106, 2136, 2145  
`\iconNameI` ..... 2106  
`\if@everyPageClose` ..... 1343, 1359, 1368  
`\if@everyPageOpen` ..... 1342, 1349, 1367  
`\if@gopro` ..... 57, 155, 1652  
`\if@useacrobat` ..... 55, 267  
`\ifaeb@ocOn` ..... 349, 350, 368, 436, 438, 439  
`\ifaeb@PageClose` ..... 1341, 1372  
`\ifaeb@PageOpen` ..... 1340, 1371  
`\ifaeb@SimpleOCOn` ..... 434, 437  
`\ifAEBP@aebpxBld@print` ..... 444, 460  
`\ifeq@proofing` .....  
.... 335, 340, 487, 491, 496, 501, 509, 525, 530  
`\ifeqforpaper` ..... 334, 486, 495, 500, 524  
`\IfFileExists` ..... 291  
`\ifHandleDblQuotes` ..... 1869, 1870  
`\ifisPDFPackage` ..... 1657, 1661  
`\ifKV@btnAnime@autopause` ..... 2407, 2439  
`\ifKV@btnAnime@autorun` ..... 2405, 2437  
`\ifKV@btnAnime@nospeedcontrol` .....  
.... 2257, 2303, 2314, 2324, 2330, 2337, 2343, 2360  
`\ifKV@btnAnime@usetworows` .....  
..... 2256, 2296, 2317, 2333, 2361  
`\ifKV@importIcons@embedonly` ..... 2075  
`\ifoptattachments` ..... 78, 1532  
`\ifoptattachmentsTaken` ..... 79, 1554  
`\ifpdf` ..... 35, 133,  
747, 788, 900, 933, 1024, 1048, 1376, 1510, 1676  
`\ifpdfmarkup` .....  
.... 145, 168, 192, 265, 274, 281, 287, 305, 696,  
736, 781, 887, 931, 1021, 1045, 1374, 1508, 1673  
`\iftexhelptoggleOff` ..... 420, 428  
`\ifxetex` ..... 36, 134,  
754, 793, 916, 934, 1027, 1051, 1378, 1511, 1678  
`\importAndSetImages` ..... 2221  
`\importDataObject` ..... 1586, 1611, 1987  
`\importIcon` ..... 1977, 2082, 2144  
`\importIcons@face` ..... 2054, 2100, 2137, 2212  
`\importIcons@fieldname` .....  
.... 2055–2057, 2095, 2097, 2131, 2133, 2207, 2209  
`\importIcons@firstpage` 2051, 2113–2115, 2142, 2147  
`\importIcons@lastpage` ..... 2052, 2116–2118, 2142  
`\importIcons@name` ..... 2046, 2069,  
2071, 2078, 2082, 2099, 2119, 2136, 2145, 2211  
`\importIcons@page` ..... 2048, 2084, 2085, 2089

`\importIcons@path` ..... 2047,  
2072, 2083, 2087, 2089, 2122, 2146, 2149, 2152  
`\importIcons@placement` .....  
.... 2049, 2075, 2091, 2092, 2127, 2128, 2203, 2204  
`\importSound` ..... 1981  
`\include` ..... 1525  
`\includecomment` ..... 502, 503, 506, 510, 511, 514  
`\includeonly` ..... 1557, 1558  
`initState` (key) ..... 17  
`initview` (key) ..... 50  
`\input` ..... 1524  
`\inputAttachmentRelatedFiles` ..... 99, 103, 162  
`\inputBtnAnimeCode` ..... 117, 119  
`\inputCommonAnimeCode` ..... 104, 108–110, 114–116  
`\InputIfFileExists` ..... 60, 69,  
86, 98, 119, 122, 125, 131, 942, 1545, 1634, 1949  
`\inputOcgAnimeCode` ..... 111, 122  
`insdljs` (option) ..... 3  
`\insertCtrlButtons` ..... 2248, 2423, 2445  
`\insertPages` ..... 1985  
`\insertPR` ..... 183, 490, 495  
`\insertPreDocAssembly` ..... 2217  
`\insertRC` ..... 181, 483  
`\ispageactions` 1345, 1350, 1353, 1360, 1363, 1367–1369  
`\isPDFPackagefalse` ..... 1661  
`\isPDFPackagetrue` ..... 1662

## J

`\JS` . 427, 428, 549, 558, 567, 809, 860, 1263, 1270,  
1278, 1284, 1290, 1296, 1351, 1355, 1361, 1365,  
1497, 1499, 1501, 1503, 1505, 1691, 1860, 2455,  
2461, 2467, 2473, 2479, 2485, 2491, 2500, 2508  
`\jsMsg` ..... 1627, 1629  
`\jsstrdotsp` ..... 1531, 1588, 1592  
`\jsstrsps` ..... 1530, 1613  
`\JStoks` ..... 1322, 1331

## K

`\kern` ..... 2252, 2315, 2331, 2343  
keys:  
`autopause` ..... 74  
`autopauseevent` ..... 74  
`autoplayevent` ..... 74  
`autorun` ..... 74  
`bgColor` ..... 32  
`border` ..... 54  
`bordercolor` ..... 55  
`center` ..... 22  
`clickAdv` ..... 33  
`controls` ..... 76



`\nFrames` ..... 543, 552, 561, 570, 2265, 2398,  
 2403, 2412, 2433, 2435, 2519, 2526, 2538, 2545  
`nFrames` (key) ..... 74  
`nopro` (option) ..... 5  
`nospeedcontrol` (key) ..... 77  
`\numWidgetsFirstRow` ..... 2243, 2244, 2359  
`\numWidgetsFirstRowV` ..... 2362, 2378  
`\numWidgetsFirstRowVI` ..... 2363, 2380

## O

`\obeyspaces` ..... 1667, 1872, 1885, 1888  
`ocgAnime` (environment) ..... 2430  
`ocganime` (option) ..... 7  
`ocgAnimeName` (key) ..... 74  
`\ocOff` ..... 458, 459  
`\offinterlineskip` ..... 2250, 2421, 2443  
`open` (key) ..... 54  
`openatpage` (key) ..... 24  
`\openout` ..... 1519, 1535, 1939  
`optattachments` (option) ..... 6  
`\optattachmentsfalse` ..... 78, 85, 161  
`\optattachmentsTakenfalse` ..... 79, 161  
`\optattachmentsTaketrue` ..... 81, 84  
`\optattachmentstrue` ..... 82  
 options:  
   `aebmlink` ..... 3  
   `aebxmp` ..... 3  
   `attachments` ..... 6  
   `attachsource` ..... 6  
   `btnanime` ..... 7  
   `dljslib` ..... 3  
   `driver` ..... 4  
   `eforms` ..... 3  
   `eq2db` ..... 3  
   `exerquiz` ..... 3  
   `gopro` ..... 6  
   `graphicxsp` ..... 3  
   `hyperref` ..... 3  
   `insdljs` ..... 3  
   `linktoattachments` ..... 6  
   `nopro` ..... 5  
   `ocganime` ..... 7  
   `optattachments` ..... 6  
   `rmannot` ..... 3  
   `useacrobat` ..... 5  
   `uselayers` ..... 5  
   `web` ..... 3

## P

`\PackageError` ..... 12, 50, 170,  
 2073, 2120, 2123, 2394, 2396, 2398, 2431, 2433  
`\PackageInfo` ..... 61, 62, 70, 71, 87, 120, 123, 126  
`\PackageWarning` ..... 88,  
 121, 124, 127, 455, 469, 481, 672, 679, 942,  
 971, 991, 1085, 1096, 1402, 1680, 2351, 2355  
`\PackageWarningNoLine` 164, 192, 195, 198, 292, 2076  
`page` (key) ..... 53, 67  
`PageDur` (key) ..... 35  
`pagelayout` (key) ..... 23  
`\PassOptionsToPackage` . 6, 41–44, 47, 48, 59, 73, 218  
`path` (key) ..... 67  
`\pathToBtnCtrlIcons` ..... 128, 2168, 2182  
`\pdfcatalog` ..... 933, 1510, 1677  
`\pdfHelp` ..... 12, 318, 336, 340  
`\pdfHelpCnt` ..... 317–320, 329–331  
`\pdfHelpi` ..... 320, 321  
`\pdfHelpIndicator` ..... 313, 323, 338  
`\pdfHelpIndicatorColor` ..... 312, 315  
`\pdflastobj` ..... 752, 791  
`\pdfmarkupfalse` ..... 153  
`\pdfmarkuptrue` ..... 146  
`\pdfobj` ..... 748, 789  
`\pdfpageattr` ..... 1025, 1026, 1049, 1050, 1376  
`\pdfpageref` ..... 905, 910, 912  
`\pdfPHProof` ..... 328, 337, 342  
`\pdfPrintHelp` ..... 329  
`\pdfPrintHelpi` ..... 331, 332  
`\pdfSP` ..... 1345  
`\pdfSpacesOff` ..... 1287, 1294  
`\placeAnimeCtrlBtnFaces` ..... 2168  
`\placeImage` ..... 2222  
`\placeImageToBtn` ..... 2201  
`placement` (key) ..... 67  
`poster` (key) ..... 74  
`\predocassemJS` . . 2044, 2081, 2094, 2141, 2206, 2220  
`\predocassemJSPlace` ..... 2126, 2130, 2154  
`preset` (key) ..... 55  
`\presets` ..... 2402, 2454, 2455,  
 2460, 2461, 2466, 2467, 2472, 2473, 2478, 2479,  
 2484, 2485, 2490, 2491, 2499, 2500, 2507, 2508  
`print` (key) ..... 16  
`printRollover` (environment) ..... 495  
`\privAddDocOpen` ..... 786, 791, 792, 796, 797  
`\prjinclude` ..... 47, 1556, 1561  
`\prjIncludeUser` ..... 1524, 1544, 1552, 1561  
`\prjinput` ..... 47, 1555, 1560  
`\prjInputUser` ..... 1524, 1542, 1551, 1560  
`\ProcessOptionsX` ..... 7, 137

`\proofRollovers` ..... 18, 184, 499, 518, 519  
`\providecommand` ..... 202, 497, 498, 719, 1959, 1963  
`\ProvidesPackage` ..... 3  
`\pubAddToDocOpen` .....  
    .... 745, 746, 752, 753, 759, 760, 890, 903, 919  
`\pushButton` ..... 324, 426, 548,  
    557, 566, 2222, 2409, 2415, 2419, 2441, 2453,  
    2459, 2465, 2471, 2477, 2483, 2489, 2496, 2504

## R

`\r` ..... 413, 414, 550, 551, 559, 560, 568, 569, 587,  
    601, 604, 606, 609, 625, 627, 2125, 2131, 2134,  
    2138, 2142, 2143, 2147, 2150, 2153, 2155, 2156,  
    2158, 2496, 2504, 2512, 2516, 2517, 2523, 2524,  
    2531, 2535, 2536, 2542, 2543, 2549, 2553, 2557  
`\raggedright` ..... 328, 520  
`\raisebox` ..... 315, 410  
`\ratio` ..... 692  
`\rawPDF` ..... 1847, 1859  
`\repeat` ..... 1450, 2417  
`req` (key) ..... 4  
`\reqpkg` ..... 291, 297, 299  
`\requiredVersionMsg` ..... 722  
`\requiredVersionMsgRedirect` ..... 722  
`\requiredVersionNumber` ..... 809, 840, 845, 855  
`\requiredVersionResult` ..... 846, 848  
`\RequirePackage` ..... 5, 8, 15–17, 94, 231, 300–302  
`\requiresVersion` ..... 722  
`\resetaddJStexHelp` ..... 415, 419, 431  
`\resetahyperDefaults` ..... 1785  
`\retnAbsPathAs` ..... 1965  
`rmannot` (option) ..... 3  
`\rollormargstring` .....  
    .... 322, 326, 333, 344, 424, 432, 521, 523, 533  
`rollover` (environment) ..... 495

## S

`\S` ..... 324, 426, 2223, 2401, 2441, 2453,  
    2459, 2465, 2471, 2477, 2483, 2489, 2498, 2506  
`\s` ..... 1530, 1531  
`\sanitize@attach` ..... 1667, 1870, 1885, 1888  
`\sbox` ..... 323, 425  
`\setahyperDefaults` ..... 1785  
`\setcounter` ..... 541  
`\setDefaultFS` ..... 938  
`\setLayoutMag` ..... 700, 714  
`\setLink` ..... 1846, 1858  
`\setPageTransition` ..... 1010  
`\setPageTransitionAt` ..... 1038  
`\setspaceBtwnPMBtns` ..... 2251, 2255

`\setUIOptions` ..... 668, 716  
`\setWindowOptions` ..... 655, 715  
`\sh@keandbake` ..... 1326, 1328  
`\shakeandbake` ..... 1058, 1060  
`showtitle` (key) ..... 22  
`\sigFieldObj` ..... 1993  
`\sigInfo` ..... 1992  
`\signatureSetSeedValue` ..... 1992  
`\signatureSign` ..... 1992  
`\small` ..... 328, 520  
`\smash` ..... 2440  
`Speed` (key) ..... 35  
`speed` (key) ..... 74  
`\stepcounter` ..... 535  
`\stringiiUnicode` ..... 1668, 1781, 1784  
`\strip@pt` ..... 693

## T

`\t` ..... 589, 591,  
    2125, 2130, 2131, 2134, 2143, 2147, 2150, 2512,  
    2516, 2523, 2531, 2535, 2542, 2549, 2553, 2557  
`\taggedPDF` ..... 719  
`\targetDictionary` ..... 1768–1770, 1849  
`\targetDictionary@cc` ..... 1770, 1783  
`\targetDictionary@cp` ..... 1769, 1782  
`\targetDictionary@pc` ..... 1768, 1780  
`\temp@exp` ..... 265, 269, 274, 276, 281, 282, 287, 288  
`\temp@expand` ..... 1580, 1581, 1601, 1602  
`\temp@expand@def` ..... 647, 652, 659, 665  
`\temp@expand@sets` ..... 713, 717  
`\termLoop` ..... 2403, 2413, 2414, 2419  
`\test@did@Print` ..... 1494, 1504  
`\test@did@Save` ..... 1492, 1500  
`\test@will@Close` ..... 1489, 1496  
`\test@will@Print` ..... 1493, 1502  
`\test@will@Save` ..... 1491, 1498  
`\texHelp` ..... 178, 421, 526, 530  
`\texHelp@i` ..... 421–423  
`\texHelpIndicator` ..... 175, 409, 425, 528  
`\texHelpIndicatorColor` ..... 408, 411  
`\texhelptoggleOfffalse` ..... 421  
`\texhelptoggleOfftrue` ..... 420, 422  
`\texorpdfstring` ..... 322, 333, 424, 523  
`\texPHProof` ..... 185, 520, 527, 531  
`\texPrintHelp` ..... 520  
`\textcolor` ..... 1845, 1857  
`textcolor` (key) ..... 55  
`\textFont` ..... 548, 557, 566, 2497, 2505  
`\texttt` ..... 328, 520  
`\theDocObject` ..... 1962, 1964



<code>\theocSeq</code>	544
<code>\thinspace</code>	313, 409
<code>\thisClosePageActionAt</code>	1306, 1307, 1312
<code>\thisOpenPageActionAt</code>	1302, 1303, 1311
<code>\thisPageAction</code>	1263, 1270, 1278, 1284, 1290, 1296
<code>timeDelay (key)</code>	32
<code>\tiny</code>	316, 412
<code>\tmp@spaceBtwnPMBtns</code>	2253, 2254
<code>\to@nilarg</code>	1392, 1583, 1605, 2057
<code>\toks</code>	1866, 1867
<code>\totalheight</code>	2249
<code>\TP</code>	2224, 2401, 2454,
	2460, 2466, 2472, 2478, 2484, 2490, 2498, 2506
<code>Trans (key)</code>	31, 35
<code>true (key)</code>	16
<code>\TU</code>	325, 548, 557, 566, 2453,
	2459, 2465, 2471, 2477, 2483, 2489, 2496, 2504
<code>\turnProofingOff</code>	498
<code>\turnProofingOn</code>	497
<code>type (key)</code>	74

## U

<code>\u</code>	1820, 1832, 1853
<code>\uCA</code>	548, 566, 2505
<code>uioptions (key)</code>	25
<code>useacrobat (option)</code>	5
<code>\usebox</code>	325, 431
<code>uselayers (option)</code>	5
<code>\usepackage</code>	297
<code>usePageTiming (key)</code>	33
<code>useTimer (key)</code>	33
<code>usetworows (key)</code>	77

## 16 Change History

v0.8f (2008/06/15)

General: Added `graphicxsp` to the list of support packages . . . . . 3

v0.8g (2008/06/30)

General: Added developer hooks to this open page events. . . . . 45

v0.8h (2008/07/19)

`\requiresVersion`: Added an optional argument to `\requiresVersion`, to give the option of only warning the user that a certain version is required; no exiting of the document is forced on the user. . . . . 25

## V

<code>\vcgBdry</code>	2304, 2324, 2338, 2421, 2444
<code>\verbatim@line</code>	1932
<code>\verbatim@out</code>	309, 1520, 1931, 1940, 1941, 1946
<code>\verbatim@processline</code>	1930
<code>\verbatim@start</code>	1933
<code>\verbatimwrite</code>	1927, 1944
<code>view (key)</code>	53
<code>viewmode (key)</code>	50
<code>\vspacectrlsep</code>	2292, 2421, 2444

## W

<code>\W</code>	2238, 2441
<code>web (option)</code>	3
<code>\widthFirstRow</code>	2228, 2368
<code>\will@Close</code>	1459, 1461, 1490, 1497
<code>\will@Print</code>	1475, 1477, 1493, 1503
<code>\will@Save</code>	1467, 1469, 1491, 1499
<code>willClose (environment)</code>	1452
<code>willPrint (environment)</code>	1452
<code>willSave (environment)</code>	1452
<code>windowoptions (key)</code>	25
<code>\write</code>	309, 1536, 1700, 1931, 1941, 1946

## X

<code>\x</code>	213, 214, 217, 218, 230, 231, 1309, 1311, 1881, 1886
<code>\xBld</code>	179, 440, 536
<code>\XKV@rm</code>	208, 215

## Y

<code>\y</code>	1310, 1312
-----------------	------------

v0.8i (2008/10/08)

General: Beginning with version 7.05, there is a `description` property. Prior to that, the value of `cname` was used for the description. I've added an additional line that sets the description equal to `cname` for 7.05 or later. 48

v1.2 (2018/04/26)

General: Added dummy package `annot-pro` . . . . . 2

v1.3f (2012/06/01)

General: Added controls for adding JS code to a rollover; used for starting and stopping an anime. . . . . 15

v1.3h (2012/11/25)		v1.7 (2015/07/21)	
General: Changed height calculation in <code>\texHelp</code>	15	<code>\insertPR</code> : Modified <code>\proofRollovers</code> command	
v1.4 (2013/02/24)		and added a number of other related	
General: Support for optional attachments	6, 46	commands. . . . .	18
v1.4b (2013/04/01)		v1.7k (2015/08/02)	
General: The <code>createicon</code> block separates the		<code>\labelName</code> : Added <code>\labelName@a</code>	59
import icon macros, doc and		Added additional logic to <code>\aeb@labelName</code>	59
pre- <code>docassembly</code> stuff. . . . .	60	v1.7k (2015/08/05)	
v1.4d (2014/02/24)		<code>\earlyAttachForPkgs</code> : Added a	
General: Within this and the next <code>\@for</code> loop,		<code>\ifisPDFPackage</code> switch . . . . .	51
added a test for the empty argument. . . . .	48	v1.7k (2015/08/06)	
v1.7 (2015/06/07)		General: Added code so that the <code>childof</code> is read	
<code>\dev@addToDocOpen</code> : Added private additional		after <code>linktoattachments</code> . . . . .	6
action . . . . .	27	v1.7l (2015/08/07)	
v1.7 (2015/06/11)		<code>\labelName</code> : Add the switch	
<code>\addToDocOpen</code> : Renamed		<code>\ifHandleDbQuotes</code> switch . . . . .	58
<code>\additionalOpenAction</code> to <code>\addToDocOpen</code> ,		<code>attachmentNames</code> : Set <code>\ifHandleDbQuotes</code> to	
this version is additive. . . . .	25	<code>true</code> . . . . .	60
v1.7 (2015/06/12)		v1.7m (2015/08/08)	
General: Added a test for <code>pdftex</code> and <code>xetex</code>	7	General: Made <code>print</code> key into a choice field	16
v1.7 (2015/06/17)		v1.7n (2015/10/14)	
General: Use <code>\firstPageSpec</code>	43, 44	<code>\afterRequirementPassedJS</code> : Inserted	
<code>addJSToPageCloseAt</code> : <code>addJSToPageOpenAt</code> can		parentheses to prevent <code>dvips</code> from breaking	
now be used in preamble . . . . .	40	lines within a string. . . . .	28
v1.7 (2015/07/01)		v1.9 (2016/03/03)	
General: Changed <code>\aeb@exiii</code> to <code>\expandafter</code> ,		General: Allow users of <code>pdftex</code> and <code>xetex</code> to	
changed <code>\aref(#1)</code> to <code>#1</code> . . . . .	56	access JavaScript related methods provided	
Changed <code>"\jobname.\@tmp"</code> to just <code>"\@tmp"</code>	48	they use Acrobat as their PDF reader. . . . .	3
Changed <code>cName</code> from <code>\jobname.\@tmp</code> to		Mark as non- <code>pdfmark</code> . . . . .	8
just <code>\@tmp</code> . . . . .	48	Modified driver check, adding	
Remove <code>\aref</code> in <code>cName</code> . . . . .	49	<code>\useacrobattrue</code> . . . . .	8
Remove <code>\aref</code> in first argument . . . . .	49	Removing <code>pdfmark</code> code for non-Distiller	
<code>\hyperextract</code> : Removed <code>\aref(#2)</code> , replaced		workflow . . . . .	30, 31
with <code>#2</code> . . . . .	58	Set up <code>\ifpdfmark</code> switch earlier . . . . .	8
<code>\hyperlink</code> : Rewrite the link to stuff, we'll use		<code>\addToDocOpen</code> : Redefine <code>\addtoDocOpen</code> only	
<code>cName</code> rather than description to link to the		for <code>pdfmarkup</code> . . . . .	26
attachments, this is how it should have been		v2.0 (2016/05/08)	
to begin with. . . . .	56	General: Attempt to extend package to include	
<code>\earlyAttachForPkgs</code> : Removed <code>\aeb@exiii</code>		more features for <code>pdftex</code> and <code>xetex</code> . Modified	
and <code>\aref()</code> , replaced with <code>\expandafter</code>		<code>\aeb@SetCatalog</code> . . . . .	31
and argument. . . . .	51	<code>\earlyAttachForPkgs</code> : Extend <code>\makePDFPackage</code>	
v1.7 (2015/07/05)		to include <code>pdftex</code> and <code>xetex</code> . . . . .	51
<code>\labelName</code> : Modified the commands <code>\labelName</code>		v2.0a (2016/05/11)	
and <code>\labelName@star</code> to gobble up any		<code>\aebPageAction</code> : Let <code>insdljs</code> handle page open for	
spaces between arguments before we switch		page 1. . . . .	42
to <code>\obeyspaces</code> . . . . .	58	v2.1 (2016/05/15)	
v1.7 (2015/07/11)		General: Changes to <code>\@setDocAction</code> for	
<code>\earlyAttachForPkgs</code> : Added		non-Distiller workflows. . . . .	45
<code>CollectionPreview</code> menu item to assure the		Moved <code>\pdfHelp</code> from <code>aeblayers</code> to <code>package</code>	
Hidden view is displayed. . . . .	50	to support non-Distiller workflows. . . . .	12

<code>\addToDocOpen</code> : Finalized changes to	Change division from postscript operation to a	
<code>\addToDocOpen</code> for non-Distiller workflows. .	tex calculation . . . . .	24
<code>\aebp@PageOpenCloseHandler</code> : Added the	Removed the generation of <code>fsprodef.js</code> from	
command <code>\aebp@PageTransHandler</code> to	the INS file. . . . .	3
handle the page transitions . . . . .	v2.4 (2016/08/03)	
Added the command	General: Removed the <code>latin1</code> option as	
<code>\aebp@PageTransHandler</code> to the shipout cmd	unworkable across some workflows . . . . .	6
<code>\AddToShipoutPicture</code> . . . . .	v2.5 (2016/08/03)	
<code>\aebPageAction</code> : Final changes to	General: Revised all these function that use	
<code>\aebPageAction</code> to include non-Distiller	<code>\theDocObject</code> . . . . .	61
workflows. . . . .	v2.5 (2016/08/31)	
<code>\dev@addToDocOpen</code> : Extended	General: Added <code>\docSaveAs</code> . . . . .	63
<code>\dev@addToDocOpen</code> to include non-Distiller	Revised <code>aeb_pro.js</code> to include	
workflows. . . . .	<code>aebDocSaveAs()</code> . . . . .	3
<code>\earlyAttachForPkgs</code> : Finalized changes in	v2.7 (2016/12/10)	
<code>\makePDFPackage</code> to accommodate	General: Incorporate the version argument into	
non-Distiller workflows. . . . .	package specification . . . . .	3
<code>\setPageTransition</code> : Finalized changes in	v2.8a (2017/03/08)	
<code>\setPageTransition</code> for non-Distiller	General: <code>\ahref@bordercolor</code> can be a named	
workflows. . . . .	color . . . . .	55
<code>\setPageTransitionAt</code> : Finalized changes in	<code>ctrlbdrycolor</code> accepts <code>xcolor</code> , if loaded . . . . .	74
<code>\setPageTransitionAt</code> for non-Distiller	v2.9 (2017/08/25)	
workflows. . . . .	General: Extracted some of the code from	
v2.10 (2021/02/07)	<code>unicode.def</code> to create a new package,	
General: The <code>@@defineJS</code> uses macros defined in	<code>forms16be</code> . This package now includes	
<code>insdljs</code> , dated 2021/02/07 or later. . . . .	<code>forms16be</code> . . . . .	3
<code>\aebPageAction</code> : Hack: redefine	require <code>forms16be</code> as needed . . . . .	12
<code>\@acrivechar@in</code> from <code>\LaTeX</code> core; this was	<code>\labelName</code> : Pass control to <code>forms16be</code> . . . . .	58
disrupting the page events. . . . .	Test for <code>attachmentNames</code> environment . . . . .	58
<code>addJSToPageCloseAt</code> : Fix a long standing	v2.9.4 (2018/12/02)	
problem with page open and close events . .	General: Require <code>aeb-comment</code> (version 3.2 of	
v2.11 (2021/04/27)	<code>comment</code> ) . . . . .	12
General: Added <code>aebCertifyInvisibleSign</code> to	v2.9.5 (2019/03/21)	
<code>aeb_pro.js</code> (Version 1.5) . . . . .	General: load <code>eforms</code> before <code>insdljs</code> . . . . .	11
v2.12 (2021/06/08)	v2.9.6 (2029/12/24)	
General: Include <code>aeb_mlnk</code> in AeB Central . . . . .	<code>\signatureSetSeedValue</code> : Use <code>\dfnJSCR</code> of	
Updated <code>aeb_pro.js</code> to Version 1.6, this	<code>insdljs</code> . . . . .	65
version includes <code>aebLaunchURL</code> . . . . .	v2.9.7 (2021/02/04)	
v2.13 (2021/06/11)	General: Updated documentation to reflect new	
General: Cannot have <code>docassembly</code> package	Acrobat security restriction, Acrobat may	
loaded . . . . .	have to be configured. . . . .	3
<code>\retnAbsPathAs</code> : Added <code>\retnAbsPathAs</code> . . . . .	v2.9a (2017/10/08)	
v2.14 (2021/06/20)	General: Added <code>\pdfHelpIndicatorColor</code> . . . . .	12
General: Require <code>acrotex-js</code> dated 2021/06/19 or	Added <code>\texHelpIndicatorColor</code> . . . . .	15
later . . . . .	v2.9b (2018/02/17)	
v2.2 (2016/07/26)	General: Added star option to <code>\texHelp</code> . . . . .	15
General: Added support for <code>luatex .85</code> . . . . .		