

rmannot: Support for FLV, SWF, and MP3 in Acrobat 9 Pro

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

processed January 30, 2021

Contents

1	Introduction	1
2	Options of this package	4
3	Preliminary Code	4
4	The <code>\rmAnnot</code> command	12
4.1	Options for <code>\rmAnnot</code>	13
4.1.1	Annot Name	13
4.1.2	Launch Settings	13
4.1.3	<code>\setRmOptions3D</code> : UI for 3D	15
4.1.4	Annot appearance options	15
4.1.5	Options for the skins	17
4.1.6	Animation Settings	18
4.1.7	Adding Resources	19
4.2	The definition of <code>\rmAnnot</code>	23
4.3	Poster Appearances	38
4.4	Support for 3D Annotations	40
4.4.1	Code from <code>movie15</code>	44
5	Index	57
6	Change History	63
1	<code><*package></code>	

1 Introduction

The `rmannot` package was written, in part, to support the AcroF_eX Graphing package; however, this package has wider application. The `rmannot` package supports the creation of rich media annotations (`RichMedia`), and the embedding of

SWF and FLV files in a PDF. SWF animations and FLV video can then be played within PDF viewed within version 9 (or later) of Adobe Reader and Acrobat.

Source material for the creation of this package is *Adobe Supplement to the ISO 32000*, June 2008. This document contains the PDF specification of rich media annotations.

Version 9 or later. Beginning with version 9, Adobe Reader and Acrobat contain a Adobe Flash Player, which will play SWF, FLV, MP3 files, and a number of other formats that need to be H.264 encoded.

After December 2020. Adobe drops its support for Flash player (SWF and FLV files) after December 2020; however, this package still works for H.264 encoded videos, including MOV, MP4, M4V, 3GP, 3G2, F4V. The only audio file format supported is MP3.

On the Topic of 3D. Here is something that I've only just come to realize: If you use the UI, and you create a 3D annotation in Acrobat, then give it a SWF as a resource, the 3D annot gets converted into a Rich Media annotation. Looking through the specification as described in the *Adobe Supplement to ISO 32000*, I determined to implement this feature, and why not since most of the structure (rich media annot) was already in place by way of my `rmannot` package. So, this version of `rmannot` supports what I'll call Rich Media 3D annotations (RM3DA).

Initially, it was not a challenge to get a 3D model to appear in a RMA created by `rmannot`, some straight forward modifications to `rmannot` were required, following ISO 32000. Looking at Alexander Grahn's very fine and brilliant `movie15` package, I saw the difficulties of defining and creating *views* through the L^AT_EX interface. With Alexander's permission, I gently lifted all the really heavy code from `movie15`, and placed it in `rmannot`. I offer up my great and humble thanks for his kindness in allowing the use of his code (characterized by commands beginning with `@MXV`).

Alexander Grahn,
he's the man!

```
2 \RequirePackage{xkeyval}
3 \RequirePackage{ifpdf}[2006/02/20]
4 \RequirePackage{ifxetex}[2006/08/21]
5 \let\rm@One=1 \let\rm@Zero=0
```

(2020/08/21) We test for non-pdfmark drivers, if present, we make minimal package definitions, define all relevant commands to display their *<text>* argument. In this way, `pdflatex`, `lualatex`, and `xelatex` can be used to preview the document, perhaps viewing the results in SumatraPDF.

```
6 \ifpdf
7   \let\RM@action\endinput
8 \else
9   \ifxetex
10    \let\RM@action\endinput
11  \else
12    \let\RM@action\relax
13  \fi
14 \fi
```

```
15 \ifx\RM@action\endinput
```

We provide definitions to all essential commands and environments to, as best as we can, give a PDF that is viewable, but with no functionality.

```
16 \RequirePackage{eforms}[2020/12/14]
17 \DeclareOptionX*{}
18 \ProcessOptionsX\relax
19 %\let\AcroVer@gobble
20 \newcommand\AcroVer[2] [] {}
21 \let\saveNamedPath@gobbletwo
22 \def\rma@edefexecute#1{\edef\rm@@temp@@exp{#1}\rm@@temp@@exp}
23 \newcommand\makePoster[3] [] {}
24 \define@key{rmAnnot}{width}{\def\rmAnnot@width{#1}}
25 \define@key{rmAnnot}{scale}{\def\rmAnnot@scale{#1}}
26 \let\rmAnnot@width\empty
27 \define@key{rmAnnot}{height}{\def\rmAnnot@height{#1}}
28 \let\rmAnnot@height\empty
29 \newcommand{\rmAnnot}[4] [] {%
30   \bgroup
31     \setlength{\dimen@}{#2}\xdef\rm@Annot@width{\the\dimen@}%
32     \setlength{\dimen@}{#3}\xdef\rm@Annot@height{\the\dimen@}%
33   \egroup
34   \rma@edefexecute{\noexpand\setkeys*{rmAnnot}{#1}}%
35   \bgroup
36   \ifx\rmAnnot@width\empty
37     \ifx\rmAnnot@height\empty
38       \else
39         \setlength{\dimen@}%
40           {\rmAnnot@height*\ratio
41             {\rm@Annot@width}{\rm@Annot@height}}%
42         \xdef\rm@Annot@width{\the\dimen@}%
43         \setlength{\dimen@}{\rmAnnot@height}%
44         \xdef\rm@Annot@height{\the\dimen@}%
45       \fi
46     \else
47       \setlength{\dimen@}%
48         {\rmAnnot@width*\ratio
49           {\rm@Annot@height}{\rm@Annot@width}}%
50       \xdef\rm@Annot@height{\the\dimen@}%
51       \setlength{\dimen@}{\rmAnnot@width}%
52       \xdef\rm@Annot@width{\the\dimen@}%
53     \fi
54   \egroup
55   {\previewOn\pushButton[\CA{Distiller required}\BC{}\BG{}
56     \S{S}\Ff{\FfReadOnly}]{btn}{\rm@Annot@width}{\rm@Annot@height}}
57 \def\setRmOptions3D#1#2{}
58 \PackageWarningNoLine{rmannot}
59 {PDF creation requires Adobe Distiller.\MessageBreak
60 Workflow is latex > dvips > distiller; otherwise,\MessageBreak
61 this package does nothing}
```

```

62 \fi
63 \RM@action % \endinput or \relax

```

2 Options of this package

use3D When the `use3D` option is invoked, the `annot3d.def` code file is input at the end of the package, and the `fp` package is loaded, is package is used to calculate the views matrices.

```

64 \DeclareOptionX{use3D}{%
65   \def\rma@input@iiidCode{\InputIfFileExists{annot3d.def}{}}{}%
66   \def\rma@requirefp{\RequirePackage[nomessages]{fp}}{}%
67 }
68 \let\rma@input@iiidCode\relax
69 \let\rma@requirefp\relax

```

Process options, there can be only one!

```
70 \ProcessOptionsX
```

We use `graphicxsp` to generate annotation appearances.

```

71 \RequirePackage{graphicxsp}
72 \rma@requirefp
73 \RequirePackage{ifthen}

```

3 Preliminary Code

A counter to track the annots as they are created.

```
74 \newcounter{rm@Cnt}
```

Some switches and markers to prevent embedding the same file multiple times.

```

75 \newif\ifrma@EmbedFile\rma@EmbedFiletrue
76 \newif\ifrma@EmbedVideoPlayer\rma@EmbedVideoPlayerfalse
77 \let\rma@isVPEEmbedded\rm@Zero
78 \newif\ifrma@EmbedAudioPlayer\rma@EmbedAudioPlayerfalse
79 \let\rma@isAPEEmbedded\rm@Zero

```

We use a utility command, taken and renamed from the `comment` package.

```
80 \def\rm@csarg#1#2{\expandafter#1\csname#2\endcsname}
```

\pathToSkins For FLV files, one of standard skins are used to control the play. We need to know where the skins are located on the system (for distiller) and where the players are. Use **\pathToSkins** to specify the location of the skins, and **\pathToPlayers**. Specifying the **\pathToSkins** also defines the path to players. Currently, the players are in a sub-folder of the `Multimedia Skins` folder. We include **\pathToPlayers** in case future releases move the players elsewhere; in this case, **\pathToPlayers** must be executed after **\pathToSkins**.

```

81 \newcommand{\pathToSkins}[1]{\begingroup
82   \Hy@unicodefalse\pdfstringdef\rma@pathToSkins{#1}%
83   \gdef\PathToSkins{\rma@pathToSkins}%

```

```

84 \gdef\rma@pathToPlayers{\rma@pathToSkins/Players}\endgroup
85 }

\AcroVer (2015/09/30) Added \AcroVer and a more intelligent method of finding the path
to skins and players. The optional argument of \AcroVer takes key words win
or mac, the default is win. Typically, the argument of \AcroVer is a number,
9, 10, 11, but beginning with the DC versions it can be 2015 (classic) or DC
(subscription).

86 \newif\ifuseWinAcrobat\useWinAcrobattrue
87 \define@choicekey{rmAcroVer}{win}{32,64}[32]{%
88 \appType{#1}\useWinAcrobattrue}
89 \define@key{rmAcroVer}{mac}[mac]{\useWinAcrobatfalse}

The syntax is \AcroVer[win|mac]{ver}

90 \def\appType#1{\def\@rgi{#1}\def\@tstii{64}%
91 \def\p@thHash{ (x86)}\ifx\@rgi\@tstii\let\p@thHash\@empty\fi
92 }
93 \def\p@thHash{ (x86)}
94 \newcommand{\AcroVer}[2][]{%
95 \def\rmDC{DC}\def\rmBeta{Beta}\def\rmArgi{#1}%
96 \def\AcrobatVer{#2}\ifx\rmArgi\@empty\else
97 \setkeys{rmAcroVer}{#1}\fi\def\@x{\string\ }%

If this is the DC version, we handle appropriately

98 \ifx\AcrobatVer\rmDC
99 \ifuseWinAcrobat
100 \edef\rmSkinPath{C:/Program Files\p@thHash/Adobe/Acrobat DC/%
101 Acrobat/Multimedia Skins}\else
102 \edef\rmSkinPath{/Applications/Adobe{\@x}Acrobat{\@x}DC/%
103 Adobe{\@x}Acrobat.app/Contents/Resources/%
104 Multimedia{\@x}Skins}\fi
105 \else\ifx\AcrobatVer\rmBeta
106 \ifuseWinAcrobat
107 \edef\rmSkinPath{C:/Program Files\p@thHash/Adobe/Acrobat Beta/%
108 Acrobat/Multimedia Skins}\else
109 \edef\rmSkinPath{/Applications/Adobe{\@x}Acrobat{\@x}Beta/%
110 Adobe{\@x}Acrobat.app/Contents/Resources/%
111 Multimedia{\@x}Skins}\fi

If \AcrobatVer is not DC or Beta it is a number. Possible values are 9, 10, 11,
2015, 2016,...

112 \else
113 \ifnum\AcrobatVer<9\relax
114 \PackageError{rmannot}{Acrobat version 9 or later
115 supports\MessageBreak rich multimedia annotations}
116 {Upgrade your Acrobat to a more recent version.}%
117 \else
118 \ifuseWinAcrobat

We are on a Windows OS machine

119 \ifnum\AcrobatVer<12\relax

```

When version is less than 12, the version numbers are decimal numbers, 9.0, 10.0, 11.0. We append ‘.0’ to the end of \AcrobatVer.

```

120         \edef\rmSkinPath{C:/Program Files (x86)/Adobe/%
121             Acrobat \AcrobatVer.0/Acrobat/Multimedia Skins}%
122     \else

```

When version is greater than 12, the version is a year 2015, etc.

```

123         \edef\rmSkinPath{C:/Program Files (x86)/Adobe/%
124             Acrobat \AcrobatVer/Acrobat/Multimedia Skins}%
125     \fi
126 \else

```

We are on a Mac OS machine

```

127     \ifnum\AcrobatVer<12\relax
128     \ifnum\AcrobatVer=9\relax

```

Special path for version 9

```

129         \edef\rmSkinPath{/Applications/%
130             Adobe{\@x}Acrobat{\@x}\AcrobatVer{\@x}Pro/%
131             Adobe{\@x}Acrobat{\@x}Pro.app/Contents/%
132             MacOS/Multimedia{\@x}Skins}%
133     \else % ver 10 or 11

```

Versions 10 and 11 are referred to using roman numerical numbers (X and XI)

```

134         \ifnum\AcrobatVer=10\relax\def\romanVer{X}\else
135         \ifnum\AcrobatVer=11\relax\def\romanVer{XI}\fi\fi
136     \edef\rmSkinPath{/Applications/%
137         Adobe{\@x}Acrobat{\@x}\romanVer{\@x}Pro/%
138         Adobe{\@x}Acrobat{\@x}Pro.app/Contents/%
139         Resources/Multimedia{\@x}Skins}%
140     \fi
141 \else

```

\AcroVer is greater than 12, it must be a year, 2015, 2016, etc.

```

142         \edef\rmSkinPath{/Applications/%
143             Adobe{\@x}Acrobat{\@x}\AcrobatVer/%
144             Adobe{\@x}Acrobat.app/Contents/Resources/%
145             Multimedia{\@x}Skins}%
146     \fi

```

```

147 \fi\fi\fi\fi
148 \expandafter\pathToSkins\expandafter{\rmSkinPath}%
149 }
150 \@onlypreamble\AcroVer
151 \AcroVer{DC}
152 \newcommand{\pathToPlayers}[1]{\pdfstringdef\rma@pathToPlayers{#1}}

```

\defineRMPath A simple command for defining paths. We use \hyper@normalise (from hyperref). Special characters are made safe to use. The command takes two arguments, the first is the control sequence of the path you want to define; the second argument is the path. For example,

```
\defineRMPath{\myURLRMFiles}{http://www.example.com/~dpspeaker/videos}
```

```

153 \newcommand{\defineRMPPath}[1]{\def\rm@ctrlName{#1}%
154 \hyper@normalise\rm@defineURLPath}
155 \def\rm@defineURLPath#1{\expandafter\xdef\rm@ctrlName{#1}}

```

`\useVideoPlayerPlus` (2010/09/29 v1.0b) Added support for the use of a video player with additional features that are nice. `VideoPlayerPlus.swf` is supplied by Joel Geraci, see his blog article at blogs.adobe.com/pdfdevjunkie/2010/03/introducing_the_video_player_p.html.

(2016/10/09) Removed support for `VideoPlayerPlus.swf`; this widget is no longer supported by the author of the widget. All functionality of the player plus widget is included in the player X widget.

```

156 \newif\ifVideoPlayerEx\VideoPlayerExfalse
157 \def\rm@VideoPlayer{VideoPlayer.swf}
158 \newcommand{\useVideoPlayerPlus}{%
159 \PackageWarning{rmanot}{The \string\useVideoPlayerPlus\space
160 is no longer supported,\MessageBreak
161 will use \string\useVideoPlayX\space instead.
162 In the future\MessageBreak
163 specify \string\useVideoPlayX}%
164 Use VideoPlayerX.swf instead.
165 \useVideoPlayerX
166 }

```

`\useVideoPlayerX` We also support the video player begin developed (and in beta) by UVSAR at <http://www.uvsar.com/projects/acrobat/videoplayerx/>.

```

166 \newcommand{\useVideoPlayerX}{\VideoPlayerExtrue
167 \def\rm@VideoPlayer{VideoPlayerX.swf}%
168 \let\useVideoPlayerPlus\relax
169 }

```

`\useVideoPlayerPlus` and `\useVideoPlayerX` allowed only in the preamble.

```

170 \@onlypreamble\useVideoPlayerPlus
171 \@onlypreamble\useVideoPlayerX

```

Javascript API for Multimedia. We present some convenience macros for controlling the multimedia players; we support two players `VideoPlayer.swf` (Adobe) and `VideoPlayerX.swf` (UVSAR). We shall refer to these three as **VPB** and **VPX**, respectively.

Core API. Valid for all supported.

```

172 \def\mmPlay{"multimedia_play"}
173 \def\mmPause{"multimedia_pause"}
174 \def\mmRewind{"multimedia_rewind"}
175 \def\mmNextCuePoint{"multimedia_nextCuePoint"}
176 \def\mmPrevCuePoint{"multimedia_prevCuePoint"}
177 \def\mmSeek{"multimedia_seek"}
178 \def\mmMute{"multimedia_mute"}
179 \def\mmVolume{"multimedia_volume"}

```

VPX API. The following are defined for **VPX**. The JavaScript API supported by UVSAR's `VideoPlayerX.swf`. Some API are more advanced and not listed, others—listed under the core above—change the return value from `void` to a non-void return. Full documentation for these is at <http://www.uvsar.com/projects/acrobat/videoplayerx/>.

```
180 \def\mmSource{"multimedia_setSource"}
181 \def\mmSkin{"multimedia_setSkin"}
182 \def\mmSkinColor{"multimedia_setSkinColor"}
183 \def\mmSeekCuePoint{"multimedia_seekCuePoint"}
184 \def\mmSkinAlpha{"multimedia_setSkinAlpha"}
185 \def\mmGetSource{"multimedia_getSource"}
186 \def\mmUseLocal{"multimedia_useLocal"}
187 \def\mmGetMetaData{"multimedia_getMetdata"}
188 \def\mmGetVideoState{"multimedia_getVideoState"}
189 \def\mmSetScaleMode{"multimedia_setScaleMode"}
190 \def\mmGetVersion{"multimedia_getVersion"}
```

Version 10.2 of **VPX**, **VPX** now recognizes API of **VideoPlayerPlus**, we therefore include `\mmSkinAutoHide`; hence, there is no reason to even use **VideoPlayerPlus**, which we now don't anymore.

```
191 \def\mmSetStageColor{"multimedia_setStageColor"}
192 \def\mmIsLooping{"multimedia_isLooping"}
193 \def\mmSkinAutoHide{"multimedia_skinAutoHide"}
```

Version 10.4 of **VPX** added the following function.

```
194 \def\mmShowLoopButton{"multimedia_showLoopButton"}
```

The **VPX** is a superset of the API of the core and **VPPlus**; there is actually no reason to use the **VPPlus** anymore.

Usage: The following example sets the source for the RMA to play.

```
var rm=this.getAnnotRichMedia(this.pageNum, "myRMA");
rm.callAS(\mmSource, "myVideo");
```

The code is valid for **VPX**, for the basic **VPB**. Extensive examples may be found on the **AeB Blog** <http://blog.acrotex.net>.

Other utility macros

```
195 \def\ps@mark{[\space}
196 \def\rma@edefexecute#1{\edef\rm@@temp@@exp{#1}\rm@@temp@@exp}
```

These label commands were taken from `movie15`, needed for comparability with of the `movie15` code being used in the 3D portion of this package.

```
197 \def\@MXV@newlabel#1#2{{%
198   \rm@csarg\xdef{#1}{#2}}}%
199 \def\@MXV@getlabelvalue#1{%
200   \rm@csarg\ifx{#1}\relax%
201     undefined%
202   \else%
203     \csname#1\endcsname%
```

```

204 \fi%
205 }%

Macro for writing labels to external *.aux file
206 \def\@MXV@labeltoaux#1#2{%
207 \@bsphack\protected@write\@auxout{}{%
208 \string\@MXV@newlabel{#1}{#2}%
209 \string\@MXV@newlabel{@#1@}{\@MXV@getlabelvalue{#1}}%
210 }\@esphack%
211 \ifthenelse{%
212 \equal{\@MXV@getlabelvalue{#1}}{undefined}\or%
213 %double check that the value hasn't changed
214 \not\equal{\@MXV@getlabelvalue{#1}}{\@MXV@getlabelvalue{@#1@}}%
215 }{%

Issue warning only once, at end of document
216 \ifthenelse{\isundefined{\@MXV@warning}}{%
217 \gdef\@MXV@warning{}%
218 \AtEndDocument{%
219 \PackageWarningNoLine{rmannot}{%
220 @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@MessageBreak
221 @@ Rerun to get object references right! @@MessageBreak
222 @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@}%
223 }%
224 }{}%
225 }{}%
226 }%

```

`\RefObjRm` holds the indirect reference to the annotation with name of #1, used with RM3DA in /GoTo3DView

```

227 \def\RefObjRm#1{\@MXV@getlabelvalue{rmAnnot_#1}}

```

Configuration File. We supply a configuration file, after the definition of `\pathToSkins` to read in the path to the skins on the local file system.

```

228 \InputIfFileExists{rmannot.cfg}{-}{}

```

`\saveNamedPath` Paths to SWF and FLV files can be saved under unique names with the command `\saveNamedPath`. The first parameter is a symbolic name (unique) and is used in the fourth argument of `\rmAnnot` and as values of the `resources` key. The path is sanitized using the `hyperref` command `\hyper@normalise`.

`\rma@useNamedPath`

When we are processing a MP3 file, we need to embed the poster image only once. the following command embeds the poster, then redefines itself to `\relax`. This macro is used in `\rma@saveNamedPath`.

```

229 \def\rma@embed@mpiii@Poster{%
230 \embedEPS[hiresbb]{ramp3poster}{ramp3poster}%
231 \global\let\rma@embed@mpiii@Poster\relax
232 }

```

`\audCtrlWd` The default legacy dimensions of the MP3 control, these are and .

```

\audCtrlHt 233 \def\audCtrlWd{613bp}\let\cntrlbrWd\audCtrlWd
234 \def\audCtrlHt{66bp}\let\cntrlbrHt\audCtrlHt

```

This is the MP3 poster image, but we'll only use it once. The command redefines itself to `\relax`.

```

235 \def\rma@set@mpiiiposter{%
236   \begin{sp@createImage}{\bbox0f{ramp3poster}}{nramp3poster}%
237     \rma@invisible
238     \ps@mark{ramp3poster} /SP pdfmark
239   \end{sp@createImage}%
240   \global\let\rma@set@mpiiiposter\relax
241 }

```

`\saveNamedPath` **Syntax:** `\saveNamedPath[<MimeType>]{<name>}{<path>}`

(12/27/10) The default for [1] was `\rma@mimetype@swf`, have now changed this to `\@empty`.

```

242 \newcommand{\saveNamedPath}[2][{}]{%
243   \edef\rm@argii{#2}\@ifundefined{rma@@#2}%
244   {\gdef\rm@thisPath{rma@@#2}}{\rma@PkEr@ii}%
245   \gdef\rm@thisMimeType{#1}%
246   \hyper@normalise\rm@saveNamedPath
247 }
248 \def\rm@saveNamedPath#1{%
249   \rm@csarg\gdef{\rm@thisPath}{#1}%
250   \rm@csarg\xdef{rma@mt@\rm@argii}{\rm@thisMimeType}%

```

We check to see if this is an MP3 file. If so, we embed the screen shot of the `AudioPlayer` controls as a default poster. The graphic file `ramp3poster.eps` needs to be on the graphics search path of L^AT_EX.

```

251   \rma@edefexecute{\noexpand\filename@parse{#1}}%
252   \@ifundefined{filename@ext}{\rma@PkEr@iii{#1}}{}%
253   \rma@edefexecute{\noexpand
254     \uppercase{\noexpand\def\noexpand\rma@tempi
255       {\filename@ext}}}

```

We define the filename with extension under a convenience command. Authors are encouraged to use this command when referencing an embedded file in the `flashvars` key. Its value should be `basename.ext`, unless part of the path is enclosed in braces, in which case, a folder may be included, for example `assets/myVideo.flv`. This value is the one that appears in the **Resources** tab of the **Edit Flash** under **Name**.

```

256   \rm@csarg\xdef{\rm@argii FileName}{%
257     \filename@base.\filename@ext}%
258   \rm@csarg\xdef{\rm@argii URL}{%
259     \filename@area\filename@base.\filename@ext}%

```

Embed the file `ramp3poster.eps`. `GraphicxSP` required.

```

260   \ifx\rma@tempi\rma@rmAnnot@type@mpiii\rma@embed@mpiii@Poster\fi
261 }

```

`\rma@useNamedPath` is used internally to access the path through its name.

```

262 \def\rma@useNamedPath#1{\@nameuse{rma@@#1}}
263 \def\rma@resource#1{\csname#1FileName\endcsname}
264 \def\rma@urlresource#1{\csname#1URL\endcsname}

```

Within an eforms widget, \Name and \urlName are defined. This is to make eforms consistent with \rmAnnot, where \Name and \urlName are let to \rma@resource and to \rma@urlresource, respectively.

```
265 \expandafter\def\expandafter\makeJSSpecials\expandafter{\makeJSSpecials
266 \let\Name\rma@resource\let\urlName\rma@urlresource
267 }
```

\rmaName Public versions of \rma@resource and \rma@urlresource. (10/18/2011) Added
\rmaUrlName public versions of \rma@resource and \rms@urlresource that have parentheses to delimit argument. This way they can be used inside the insDLJS env.

```
268 \let\rmaName\rma@resource
269 \def\rmaNameP(#1){\rma@resource{#1}}
270 \let\rmaUrlName\rma@urlresource
271 \def\rmaUrlNameP(#1){\rma@urlresource{#1}}
```

Supported Extensions & Mime Types. A couple of text macros, used to compare with the extensions provided by the document author. Currently, we support SWF, FLV, MP3 files, and F4V files (Acrobat 10 or later).

(2011/11/03) Added support for MP4, M4V, MOV, 3GP, 3G2, all require H.264 encoding; if these files do not have H.264 encoding, rmannot/distiller will embed anyway, but the video will not play. (2011/12/10) Added U3D and PRC to the list of file types supported.

```
272 \def\getargsiii#1#2#3{\def\aeb@argi{#1}\def\aeb@argii{#2}%
273 \def\aeb@argiii{#3}}
274 \@tfor\rma@data:={{uiiid}}{U3D}}{model/u3d}}>{{prc}}{PRC}}{model/prc}}%
275 {{swf}}{SWF}}{application/x-shockwave-flash}}}%
276 {{flv}}{FLV}}{video/x-flv}}>{{fiv}}{F4V}}{video/mp4}}}%
277 {{mpiv}}{MP4}}{video/mp4}}>{{mivV}}{M4V}}{video/x-m4v}}}%
278 {{mov}}{MOV}}{video/quicktime}}>{{iiiGP}}{3GP}}{video/3gpp}}}%
279 {{iiiGii}}{3G2}}{video/3gpp2}}>{{mpiii}}{MP3}}{audio/x-mp3}}}\do{%
280 \expandafter\getargsiii\rma@data
281 \rm@csarg\edef\rma@rmAnnot@type@\aeb@argi}{\aeb@argii}
282 \rm@csarg\edef\rma@mimetype@\aeb@argi}{\aeb@argiii}
283 }
```

VideoPlayer Symbolic name of the video player and audio players.

AudioPlayer We delay the definitions of VideoPlayer, AudioPlayer, and the skins until the beginning of the document, if the user can specify \useVideoPlayerX in the preamble, \rma@VideoPlayer will expand to VideoPlayerX.swf. Also, the new command \AcroVer may be executed in the preamble to set the path to the skins, so we'll delay all \saveNamedPath definitions to the beginning of the document.

```
284 \def\rm@SkinsAndPlayerPaths{%
285 \saveNamedPath{VideoPlayer}{\PathToSkins/Players/\rma@VideoPlayer}%
286 \saveNamedPath{AudioPlayer}{\PathToSkins/Players/AudioPlayer.swf}%
```

skin1 We predefine the seven skins, these should also be used as resources of \rmAnnot
all when either VideoPlayerPlus or VideoPlayerX is used, and when the skins are
skin2 to be changed dynamically.

```
skin3
skin4
skin5
skin6
skin7
```

```

287 \saveNamedPath{skin1}{\PathToSkins/SkinOverAllNoFullNoCaption.swf}%
288 \saveNamedPath{all}{\PathToSkins/SkinOverAllNoFullNoCaption.swf}%
289 \saveNamedPath{skin2}{%
290     {\PathToSkins/SkinOverAllNoVolNoCaptionNoFull.swf}%
291 \saveNamedPath{skin3}{\PathToSkins/SkinOverPlay.swf}%
292 \saveNamedPath{skin4}{\PathToSkins/SkinOverPlayMute.swf}%
293 \saveNamedPath{skin5}{\PathToSkins/SkinOverPlaySeekMute.swf}%
294 \saveNamedPath{skin6}{\PathToSkins/SkinOverPlaySeekStop.swf}%
295 \saveNamedPath{skin7}{\PathToSkins/SkinOverPlayStopSeekMuteVol.swf}%
296 }

```

Now, make all these path definitions at the beginning of the document.

```
297 \AtBeginDocument{\rm@SkinsAndPlayerPaths}
```

none Two special convenience definitions. We make definitions so that `\Name{none}`
noChange and `\urlName{none}` expand to `\empty`; `\Name{noChange}` and `\urlName{none}`
both expand to the string `noChange`.

```

298 \@namedef{noneFileName}{}
299 \@namedef{noneURL}{}
300 \@namedef{noChangeFileName}{noChange}
301 \@namedef{noChange}{noChange}

```

The following is a convenience text macro, this string appears repeatedly throughout this file.

```
302 \def\rma@ANT{\rmAssetsNameTree-\therm@Cnt}
```

4 The `\rmAnnot` command

The `\rmAnnot` command creates a rich media annotation (`AnnotRichMedia`). Currently, this package supports SWF, FLV, F4V, MP4, M4V, MOV, 3GP, 3G2, and MP3 files. Normally, SWF files are applications that contains their own navigation controls; the FLV and F4V files are played by a `VideoPlayer.swf` file, shipped with Acrobat 9 Pro, and a controlling skin, also shipped with Acrobat. The `\rmannot` tries to support most of the features available through the user interface.

Acrobat 9 Pro supports FLV, SWF, and MP3 files as well, Acrobat 9 Pro Extension supports other video formats, by first converting that format to FLV and embedding them in the document. Adobe Flash (CS5) Professional can save a movie as SWF, and the included utility Adobe Flash Video Encoder can convert movie files to the FLV or F4V format.

Beginning with version 9.2, the direct inclusion of videos with extensions of MP4, M4V, MOV, 3GP, 3G2 requires that the files use the H.264 codec.

3D Support. (2011/12/10) Added U3D and PRC to the list of file types supported; when one of these files appears in #4 of `\rmAnnot` a RM3DA is created.

Certain features of 3D, specifically **3D measurement** and **3D commenting**, as well the **Select Model**, **Select Face**, and **Select 3 Points** found in the **Camera Properties** dialog box, may or may not work, depending on the information is the 3D file. For example `dice.u3d` cannot be measured, unless it is imported by

Acrobat, which apparently parses the file to obtain the required information for measuring.

4.1 Options for `\rmAnnot`

The `xkeyval` package is used to develop options for the `\rmAnnot` command.

4.1.1 Annot Name

name Use the name key to specify the annotation name. If not specified, this package generates the name `aebRM\therm@Cnt`.

```
303 \define@key{rmAnnot}{name}[aebRM\therm@Cnt]{\def\rma@Annot@name{#1}}
```

4.1.2 Launch Settings

enabled This option determines how the annot is activated, there are three possible values: `onclick` (activated when user clicks on the annot, or by user script); `pageopen` (activated when the page is opened); `pagevisible` (activated when the page becomes visible).

```
304 \define@choicekey+{rmAnnot}{enabled}[\val\nr]%
305 {onclick,pageopen,pagevisible}[onclick]{%
306 \ifcase\nr\relax
307 \def\rma@rmAnnot@enabled{/XA}\or
308 \def\rma@rmAnnot@enabled{/PO}\or
309 \def\rma@rmAnnot@enabled{/PV}\fi
310 }{\PackageWarning{rmannot}{Bad choice for enabled, permissible values
311 are onclick, pageopen and pagevisible. Try again}}
```

deactivated This option determines how the annot is de-activated, there are three possible values: `onclick` (de-activated by user script or by right-clicking on the annot and choosing Disable Content); `pageclose` (de-activated when the page is closed); `pageinvisible` (de-activated when the page becomes invisible).

```
312 \define@choicekey+{rmAnnot}{deactivated}[\val\nr]%
313 {onclick,pageclose,pageinvisible}[onclick]{%
314 \ifcase\nr\relax
315 \def\rma@rmAnnot@deactivated{/XD}\or
316 \def\rma@rmAnnot@deactivated{/PC}\or
317 \def\rma@rmAnnot@deactivated{/PI}\fi
318 }{\PackageWarning{rmannot}{Bad choice for deactivated, permissible
319 values are onclick, pageclose and pageinvisible. Try again}}
```

windowed When this boolean option is set to true (or is just included in the option list), the video is viewed in a floating window. The default is to view the video within the annot.

```
320 \define@boolkey{rmAnnot}{windowed}[true]{}
```

When the rich media annotation appears as a floating window, the initial dimensions and position of that window can be set by the following key-value pairs. Use `\setWindowDimPos` to set all these values.

```

321 \define@key{winDimPos}{width}{\def\rma@winDimPos@width{#1}}
322 \define@key{winDimPos}{width}{default}[288]%
323 {\def\rma@winDimPos@width@def{#1}}
324 \define@key{winDimPos}{width}{max}[576]%
325 {\def\rma@winDimPos@width@max{#1}}
326 \define@key{winDimPos}{width}{min}[72]%
327 {\def\rma@winDimPos@width@min{#1}}
328 \define@key{winDimPos}{height}{\def\rma@winDimPos@height{#1}}
329 \define@key{winDimPos}{height}{default}[216]%
330 {\def\rma@winDimPos@height@def{#1}}
331 \define@key{winDimPos}{height}{max}[432]%
332 {\def\rma@winDimPos@height@max{#1}}
333 \define@key{winDimPos}{height}{min}[72]%
334 {\def\rma@winDimPos@height@min{#1}}
335 \define@key{winDimPos}{position}{\def\rma@winDimPos@position{#1}}
336 \define@choicekey+{winDimPos}{halign}{\val\nr}%
337 {near,center,far}[far]{%
338 \ifcase\nr\relax
339 \def\rma@winDimPos@halign{/Near}\or
340 \def\rma@winDimPos@halign{/Center}\or
341 \def\rma@winDimPos@halign{/Far}\fi
342 }{}
343 \define@choicekey+{winDimPos}{valign}{\val\nr}%
344 {near,center,far}[near]{%
345 \ifcase\nr\relax
346 \def\rma@winDimPos@valign{/Near}\or
347 \def\rma@winDimPos@valign{/Center}\or
348 \def\rma@winDimPos@valign{/Far}\fi
349 }{}
350 \define@key{winDimPos}{hoffset}[18]%
351 {\def\rma@winDimPos@hoffset{#1}}
352 \define@key{winDimPos}{voffset}[18]%
353 {\def\rma@winDimPos@voffset{#1}}

```

`\setWindowDimPos` When the window is floating (`windowed=true`) there are a number of parameters that govern the dimensions and position on page. These key-value pairs are not set up as optional arguments of `\rmAnnot`. Set them in vertical mode, for the next rich media annotation.

```

354 \providecommand{\setWindowDimPos}[1]{%
355 \setkeys{winDimPos}{#1}%
356 \edef\temp@expand@sets{%
357 \noexpand\setkeys{winDimPos}{width}{\rma@winDimPos@width}%
358 \noexpand\setkeys{winDimPos}{height}{\rma@winDimPos@height}%
359 \noexpand\setkeys{winDimPos}{position}{\rma@winDimPos@position}%
360 }\temp@expand@sets
361 }

```

`\resetWindowDimPos` The command `\resetWindowDimPos` resets the window parameters to their default values.

```

362 \providecommand{\resetWindowDimPos}{%
363   \setWindowDimPos{width={default,max,min},height={default,max,min},
364     position={halign,valign,hoffset,voffset}}}%
365 }
366 \resetWindowDimPos

```

`url` Use the `url` key, a boolean, to indicate that the path to the file is a URL.

```

367 \define@boolkey{rmAnnot}{url}[true]{}

```

4.1.3 \setRmOptions3D: UI for 3D

`\setRmOptions3D` 3D options for the annot with name of #1; the options are passed by #2. This supports the 3D tab of the user interface for the Edit 3D dialog box.

Proposed syntax:

```

\setRmOptions3D{myDice}
{
  3DOptions={\options-from-movie15},
  3DResources={
    none={rName=<name1>},...,
    foreground={rName=<name2>,flashvars=<vars>},...,
    background={rName=<name3>,flashvars=<vars>},...,
    material={rName=<name4>,
              mName=<materialName>,flashvars=<vars>},...
  }
}

```

When the `\rmAnnot` does not take a 3D file as its 4th argument, the above options are ignored.

```

368 \def\setRmOptions3D#1#2{\rm@csarg\xdef{#1_3DOPTS}{#2}}

```

4.1.4 Annot appearance options

`borderwidth` The width of the border of the annot. Possible values are `none` (the default), `thin`, `medium`, and `thick`.

```

369 \define@choicekey+{rmAnnot}{borderwidth}[\val\nr]%
370 {none,thin,medium,thick}[none]{}%
371 \ifcase\nr\relax
372   \def\rma@rmAnnot@borderwidth{0}\or
373   \def\rma@rmAnnot@borderwidth{1}\or
374   \def\rma@rmAnnot@borderwidth{3}\or
375   \def\rma@rmAnnot@borderwidth{5}\fi
376 }\PackageWarning{rmannot}{Bad choice for borderwidth, permissible
377   values are none,thin,medium,and thick. Try again}}

```

`poster` The name of a embedded graphic to be used as a poster for the video.
`posternote` If the `poster` key is not specified, a substitute poster will be generated, see the definition of `\defaultPoster`. This default poster has a little message, or note,

in the lower left corner. The default message is an advertisement for AcroTeX followed by the words Flash, Video, or MP3, depending on the file type.

```

378 \define@key{rmAnnot}{poster}[]{\def\rma@rmAnnot@poster{#1}}
379 \define@key{rmAnnot}{posternote}[AcroTeX \rma@poster@descrip]%
380 {\def\rma@posternote{#1}}
381 \define@boolkey{rmAnnot}{defaultposter}[true]{}

invisible (2010/09/29 v1.0a) When the invisible option is used and there is no poster
option, the poster is transparent. This makes it useful when viewing the video in
a window, and you want to hide the annot in an obscure corner of the page (or
under a form field). In this case, the video/audio is played by JavaScript.

382 \define@key{rmAnnot}{invisible}[]%
383 {\def\rma@invisible{\ps@mark/ca 0/SetTransparency pdfmark }}
384 \let\rma@invisible\@empty

transparentBG This option is available only for SWF files. Set the background to transparent.

385 \define@boolkey{rmAnnot}{transparentBG}[true]{%
386 \ifKV@rmAnnot@transparentBG
387 \def\rma@rmAnnot@transparent{true}\else
388 \def\rma@rmAnnot@transparent{false}\fi
389 }

width We attempt to resize the annot according to the width or height. Resize propor-
height tionally. Only the first one of these two keys is obeyed, never both.

390 \define@key{rmAnnot}{width}{\def\rma@rmAnnot@width{#1}}
391 \let\rma@rmAnnot@width\@empty
392 \define@key{rmAnnot}{height}{\def\rma@rmAnnot@height{#1}}
393 \let\rma@rmAnnot@height\@empty

scale We attempt to resize the RMA according to the scale factor provided. The scale
key is only obeyed if no width or height key is specified.

394 \define@key{rmAnnot}{scale}{\def\rma@rmAnnot@scale{#1}}
395 \let\rma@rmAnnot@scale\@empty

toolbar These are keys concern 3D annots. toolbar is a Boolean, which if true (the
modeltree default), causes the 3D toolbar to appear when the annot is activated. If
toolbar=false, the toolbar does not appear when the annotation is activated.

396 \define@boolkey{rmAnnot}{toolbar}[true]{%
397 \ifKV@rmAnnot@toolbar
398 \def\rma@rmAnnot@toolbar{true}\else
399 \def\rma@rmAnnot@toolbar{false}\fi
400 }

modeltree modeltree is a Boolean, which if true causes the Model Tree as viewed in the
Navigation Pane. The default is false, the Model Tree is not displayed when
the annotation is activated.

401 \define@boolkey{rmAnnot}{modeltree}[true]{%
402 \ifKV@rmAnnot@modeltree
403 \def\rma@rmAnnot@modeltree{true}\else
404 \def\rma@rmAnnot@modeltree{false}\fi
405 }

```

passcontext This option is available only for SWF files. SWF file developers can select this option to replace the Acrobat context menu with the context menu of the originating SWF file. When the user right-clicks the SWF file, the available options are from the originating file.

Pass right-click context to Flash. Should be used only if there is a way of deactivating the annotation, perhaps through JavaScript.

```
406 \define@boolkey{rmAnnot}{passcontext}[true]{%
407   \ifKV@rmAnnot@passcontext
408   \def\rma@rmAnnot@PassContextClick{true}\else
409   \def\rma@rmAnnot@PassContextClick{false}\fi
410 }
```

4.1.5 Options for the skins

Skins are used with video files.

skin When playing a supported video file, various skins can be used. I've labeled them **skin1–skin7**, **all**, and **none**. The names of the SWF files describe each skin.

Note: By experimenting with the UI, it is apparent that one cannot have different controls for the same video file.

```
411 \define@choicekey+{rmAnnot}{skin}[\val\nr]%
412 {none,all,skin1,skin2,skin3,skin4,skin5,skin6,skin7}[skin1]{%
413   \edef\rma@skinName{#1}% 2011/10/18 changed from number to name
414   \ifcase\nr\relax
415     \let\rma@rmAnnot@Skin\empty\or % none
416     \def\rma@skinName{skin1}%
417     \def\rma@rmAnnot@Skin{SkinOverAllNoFullNoCaption.swf}\or % all
418     \def\rma@rmAnnot@Skin{SkinOverAllNoFullNoCaption.swf}\or % 1
419     \def\rma@rmAnnot@Skin{SkinOverAllNoVolNoCaptionNoFull.swf}\or % 2
420     \def\rma@rmAnnot@Skin{SkinOverPlay.swf}\or % 3
421     \def\rma@rmAnnot@Skin{SkinOverPlayMute.swf}\or % 4
422     \def\rma@rmAnnot@Skin{SkinOverPlaySeekMute.swf}\or % 5
423     \def\rma@rmAnnot@Skin{SkinOverPlaySeekStop.swf}\or % 6
424     \def\rma@rmAnnot@Skin{SkinOverPlayStopSeekMuteVol.swf}\fi % 7
425 }{%
426   \@ifundefined{rma@@#1}{%
427     \PackageWarning{rmannot}{Bad choice for 'skin,' permissible
428     values are none, all, skin1--skin7, or a custom skin already
429     defined. Try again}}{%
```

If a value of **skin** is not one of the defaults, we allow the user to define a skin, name and path to a skin SWF must be declared using **\saveNamedPath**.

```
430   \PackageWarning{rmannot}{Recording new skin, '#1'}%
431   \edef\rma@skinName{#1}%
432   \edef\rma@rmAnnot@Skin{\csname#1FileName\endcsname}%
433   \rm@csarg\let{embedSkin#1}\rm@One
434 }%
435 }
```

We use `\let\csname embedSkin<skinName>\endcsname` determining to embed or not to embed a particular skin. We initialize these markers to 1 (embed). After each annot, the marker corresponding to the skin used is set to 0; hence we will not embed it again.

```
436 \@tfor\rma@arg:={skin0}{skin1}{skin2}{skin3}{skin4}%
437   {skin5}{skin6}{skin7}\do{%
438   \rm@csarg\let{embedSkin\rma@arg}\rm@One
439 }
```

skinAutoHide A Boolean key that determines if the skin automatically hides itself when the mouse pointer is removed from the annot. The default is `true`.

```
440 \define@boolkey{rmAnnot}{skinAutoHide}[true]{%
441   \ifKV@rmAnnot@skinAutoHide
442     \def\rma@skinAutoHide{true}\else
443     \def\rma@skinAutoHide{false}\fi
444 }
```

skinBGColor The color of the skin, represented as a hex number, the default is `0x5F5F5F`.

```
445 \define@key{rmAnnot}{skinBGColor}[0x5F5F5F]{%
446   {\def\rma@skinBGColor{#1}}}
```

skinBGAlpha The alpha for the skin, a number between 0 and 1. The default is 0.75.

```
447 \define@key{rmAnnot}{skinBGAlpha}[0.75]{%
448   {\def\rma@skinBGAlpha{#1}}}
```

volume The initial volume of the audio track. Values range from 0 (muted) to 1. The default is 1.0. The volume may be adjusted by the user at run time, if the selected skin has a volume control, or by a JavaScript control.

```
449 \define@key{rmAnnot}{volume}[1.00]{%
450   {\def\rma@rmAnnot@volume{#1}}}
```

4.1.6 Animation Settings

speed Description quoted from the *Adobe Suppl. Doc*: A positive number specifying the speed to be used when running the animation. A value greater than one shortens the time it takes to play the animation, or effectively speeds up the animation.

```
451 \define@key{rmAnnot}{speed}[1]{%
452   \def\rma@rmAnnot@speed{#1}%
453 }
```

playcount Description quoted from the *Adobe Suppl. Doc*: An integer specifying the play count for this animation style. A nonnegative integer represents the number of times the animation is played. A negative integer indicates that the animation is infinitely repeated. The default is -1.

```
454 \define@key{rmAnnot}{playcount}[-1]{%
455   \def\rma@rmAnnot@playcount{#1}%
456 }
```

4.1.7 Adding Resources

resources Some SWF files require other files to run. The resources key allows you to list all files that are required to run the SWF file. Currently, the additional resources are other SWF files only. Items must be specified using `\useNamedPath` command, and is a lists of delimited by braces

```
resources={mySWF1,mySWF2}
```

where mySWF1 and mySWF2 are defined by `\saveNamedPath`.

```
457 \newtoks\rma@toks\rma@toks={}
458 \newcount\rma@nResources
459 \newif\ifrma@isiid\rma@isiidfalse
460 \define@key{rmAnnot}{resources}[]{\rma@toks={}\rma@nResources=0\relax
461   \ifrma@isiid\let\rma@next\relax
462   \else\def\rma@next{\rma@proc@resources{#1}}\fi\rma@next
463 }
```

When we are doing 3D, we don't process resources through `rmAnnot` family. When we are processing 3D resources, the files are entered through a separate command `\setRmOptions3D`

```
464 \newcommand{\rma@proc@resources}[1]{%
465   \def\rma@rmAnnot@resources{#1}%
466   \ifx\rma@rmAnnot@resources\@empty\let\rma@addResources\@empty
467   \let\rma@addFileSpecs\@empty\else
```

We process resources when there are some to process :-)

```
468   \@for\rma@arg:=\rma@rmAnnot@resources\do{%
469     \advance\rma@nResources1\relax
470     \rma@edefexecute{\noexpand
471       \filename@parse{\rma@useNamedPath{\rma@arg}}}%
472     \@ifundefined{filename@ext}{%
473       \rma@PkEr@iii{\rma@useNamedPath{\rma@arg}}}{%
474       \edef\rma@fs@expand{\rmFileStrm\rma@arg}%
475       \@ifundefined{\rma@fs@expand}{%
```

If this resource has not been used, we define it, and give it a indirect reference:

```
rmfstream\therm@Cnt-\the\rma@nResources
```

```
476   \rm@csarg\xdef{\rma@fs@expand}%
477   {rmfstream\therm@Cnt-\the\rma@nResources}%
478   \def\rma@embed{1}}{\def\rma@embed{0}}%
```

Add this file to our token list of all resources.

```
479   \edef\rma@tmp@exp{\the\rma@toks%
480     \noexpand\\{\the\rma@nResources}%
481     {\filename@area}{\filename@base.\filename@ext}%
482     {\rma@embed}{\csname\rma@fs@expand\endcsname}%
```

(12/27/10) changed the expansion to `\expandafter\noexpand` to leave a token as as this argument. Later, in `\rm@appendFileSpecs` we test this token against `\@empty`. This argument is #6 in `\rm@appendFileSpecs`.

```

483         {\rm@csarg\noexpand{\rma@mt@\rma@arg}}}%
484         \rma@toks=\expandafter{\rma@tmp@exp}%
485     }% do
486     \let\\\rm@appendNameTree
487     \expandafter\xdef\expandafter\rma@addResources%
488         \expandafter{\the\rma@toks}%
489     \let\\\rm@appendFileSpecs
490     \expandafter\xdef\expandafter\rma@addFileSpecs%
491         \expandafter{\the\rma@toks}%
492 \fi
493 }

```

The two commands `\rm@appendNameTree` and `\rm@appendFileSpecs` take the same six parameters, here they are

- #1 The resource number, typically `\the\rma@nResources`
- #2 `\filename@area`
- #3 `\filename@base.\filename@ext`
- #4 The value of `\rma@embed`, 1 to embed, 0 for not embedding file has already been embedded.
- #5 The indirect reference to the file stream
- #6 The mime type of this resource

`\rm@appendNameTree` This is used with the token list `\rma@toks`, before this list is expanded, `\\` is `\let` to `\rm@appendNameTree`. We create a name tree of additional resources specified by the `resources` key. This is saved in a macro `\rma@addFileSpecs` to be used later.

```

494 \def\rm@appendNameTree#1#2#3#4#5#6{%
495     \ps@mark{\rma@ANT} (#3) /APPEND pdfmark^^J%
496     \ps@mark{\rma@ANT} {rmfilespec\therm@Cnt-#1}/APPEND pdfmark^^J%
497 }

```

`\rm@appendFileSpecs` This is used with the token list `\rma@toks`, before this list is expanded, `\\` is `\let` to `\rm@appendFileSpecs`. We create a set of file specs and file streams, if needed, for each of the resources. This is saved in a macro `\rma@addResources` to be used later.

```

498 \def\rm@appendFileSpecs#1#2#3#4#5#6{%

```

File specs. If things work out well, `/F {#5}` is an indirect reference to the file stream, whether it is a new one that has never been embedded, or is one that has already been embedded.

```

499     \ps@mark/_objdef {rmfilespec\therm@Cnt-#1}/type/dict/OBJ pdfmark^^J%
500     \ps@mark{rmfilespec\noexpand\therm@Cnt-#1} <<%
501         /F(#3)%
502         /UF (#3)%
503         /EF <</F {#5}>>%
504         /Type/Filespec%
505     >>/PUT pdfmark^^J%

```

File stream: We only embed if #4 is equal to 1, if it is 0, this particular file has already been embedded.

```

506 \if#41
507 \ps@mark/_objdef {#5} /type/stream/OBJ pdfmark^^J%
508 \ps@mark{#5} (#2#3) (r) file /PUT pdfmark^^J%
509 \ps@mark{#5}<<%
510 /Type/EmbeddedFile%
(12/27/10) The /Subtype key is optional, so we make it optional if #6 is empty.
511 \ifx#6\@empty\else
512 /Subtype(#6)%
513 \fi
514 >>/PUT pdfmark^^J%
515 \ps@mark{#5} /CLOSE pdfmark^^J%
516 \fi
517 }

```

flashvars Flash variables to pass to a SWF file.

```

518 \define@key{rmAnnot}{flashvars}[] {%
519 \def\rma@rmAnnot@flashvars{#1}%
520 }

```

cuepoints Populate the CuePoint dictionary of the Params dictionary. This one is tricky. There are two types of cue points, Navigation and Event. Navigation corresponds to cue points that have been encoded onto the Flash media; Event corresponds a time value on the media, and is not, I believe, encoded on the media. We need the type (Navigation or Event), the Name, the Time and the action (A).

Event cue points are used to trigger ActionScript methods when the cue point is reached, and let you synchronize the video playback to other events within the Flash presentation.

Navigation cue points are used for navigation and seeking, and to trigger ActionScript methods when the cue point is reached. Embedding a navigation cue point inserts a keyframe at that point in the video clip to enable viewers to seek to that place in the video.

```

521 \define@key{rmAnnot}{cuepoints}[] {%
522 \edef\rma@rmAnnot@cuepoints{#1}%
523 \ifx\rma@rmAnnot@cuepoints\@empty\else
524 \rma@nResources=0\relax% dps
525 \def\rma@array@hold{}\def\rma@dict@hold{%
526 \@for\arg:=\rma@rmAnnot@cuepoints\do{%
527 \advance\rma@nResources1\relax
528 \rma@edefexecute{\noexpand\setkeys{rmCuePt}{\arg}}%
529 % need to build the array of indirect references,
530 \edef\rma@array@hold{\rma@array@hold\space
531 {rmCuePoints\therm@Cnt-\the\rma@nResources}}%
532 % and the code for the cue point dictionary
533 \edef\rma@dict@hold{\rma@dict@hold
534 \ps@mark/_objdef %
535 {rmCuePoints\therm@Cnt-\the\rma@nResources}%

```

```

536      /type/dict/OBJ pdfmark^^J%
537      \ps@mark{\rmCuePoints\therm@Cnt-\the\rm@nResources} <<
538      /Type/CuePoint
539 \ifx\rm@rmCuePt@name\@empty
540      /Name (RMACP \the\rm@nResources)
541 \else
542      /Name (\rm@rmCuePt@name)
543 \fi
544      /Subtype \rm@rmCuePt@type
545      /Time \rm@rmCuePt@time
546 \ifx\rm@rmCuePt@action\@empty\else
547      /A << /Type/Action\JS{\rm@rmCuePt@action} >>
548 \fi
549      >> /PUT pdfmark^^J%
550  }%
551  }% end of \@for
552 \fi
553 }

```

type (cuepoints) These are the key-value pairs used for defining cue points, they are type, name,
 name (cuepoints) time, and action. These keys are entered as a value of the cuepoints key, like
 time (cuepoints) so,
 action (cuepoints)

```

\newcommand{\myCuePoints}{%
  {type=nav,name=Chapter1,time=0,action={console.println("Chapter1")}},%
  {type=nav,name=Chapter2,time=1883,action={console.println("Chapter2")}},%
  {type=nav,name=Chapter3,time=5197,action={console.println("Chapter3")}},%
  {type=nav,name=Chapter4,time=6817,action={console.println("Chapter4")}},%
  {type=nav,name=Chapter5,time=9114,action={console.println("Chapter6")}},%
  {type=nav,name=Chapter6,time=12712,action={console.println("Chapter6")}}
}

```

Note the use of the comment symbol % following at the end of each line. We define the cue points using a command, then pass it to \rmAnnot, like so,

```
\rmAnnot[cuepoints={\myCuePoints}]{320bp}{240bp}{sample}
```

Note the parentheses around \myCuePoints, the command expands to contain commas, so the rules of xkeyval say to enclose in parentheses.

The structure generated in the PDF file seems to be correct, but we get JavaScript errors. Could this be a bug in Acrobat? For now, this a temporary feature.

```

554 \define@choicekey+{\rmCuePt}{type}{\val\nr}%
555   {event,nav}[nav]{%
556   \ifcase\nr\relax
557     \def\rm@rmCuePt@type{/Event}\or
558     \def\rm@rmCuePt@type{/Navigation}\fi
559 }{\PackageWarning{\rmannot}{Bad choice for type, permissible values
560   are event and nav. Try again}}
561 \define@key{\rmCuePt}{name}[]{}

```

```

562 \def\rma@rmCuePt@name{#1}%
563 }
564 \define@key{rmCuePt}{time}[0]{%
565 \def\rma@rmCuePt@time{#1}%
566 }
567 \define@key{rmCuePt}{action}[]{}%
568 \def\rma@rmCuePt@action{#1}%
569 }

```

Default values of key-values pairs. We set the default values of all the keys on startup.

```

570 \setkeys{rmAnnot}{name,url=false,enabled,deactivated,borderwidth,%
571 windowed=false,poster,posternote,skin,skinAutoHide,skinBGColor,%
572 skinBGAlpha,volume,speed,playcount,resources,flashvars,%
573 transparentBG=false,passcontext=false,cuepoints,toolbar,%
574 modeltree=false}

```

4.2 The definition of \rmAnnot

The following two utility commands are used in the definition of \rmAnnot to record the embedding of the video/audio player, and to set switches to prevent the re-embedding of the player again.

```

575 \def\rma@recordVideoPlayer{%
576 \ifrma@EmbedVideoPlayer
577 \global\let\rma@isVPEmbedded\rm@One\global\rma@EmbedVideoPlayerfalse
578 \else
579 \if\rma@isVPEmbedded\rm@Zero
580 \global\rma@EmbedVideoPlayertrue\fi
581 \fi
582 }
583 \def\rma@recordAudioPlayer{%
584 \ifrma@EmbedAudioPlayer
585 \global\let\rma@isAPEmbedded\rm@One\global\rma@EmbedAudioPlayerfalse
586 \else
587 \if\rma@isAPEmbedded\rm@Zero
588 \global\rma@EmbedAudioPlayertrue\fi

```

If the file is MP3, and there is no poster defined for it, we use the default MP3 poster, a screen shot of the AudioPlayer controls.

```

589 \ifx\rma@rmAnnot@poster\empty\rma@set@mpiiiposter\fi
590 % \rma@set@mpiiiposter
591 \fi
592 }

```

A switch to indicate success or failure when looking for a Video extension. Used in \rmAnnot.

```

593 \newif\if@FndSuppExt \@FndSuppExtfalse

```

\rmAnnot The \rmAnnot command creates a rich media annotation. This package supports file types SWF, FLV, F4V, MP4, M4V, MOV, 3GP, 3G2, and MP3, the latter being a

sound format. The command takes four parameters, the first optional, the others required.

[#1]: Insert key-value pairs for changing the appearance, launch, and control settings

#2: The width of the annot

#3: The height of the annot

#4: The name of the file to play (the name is defined by the `\saveNamedPath` command)

```
594 \newcommand{\rmAnnot}[4][\begingroup
595   \PMPV{#4}%
```

We `\let \Name` to `\rma@resource`. The author can then refer to the name of the resource within, for example, the `flashvars` key.

```
596   \let\Name\rma@resource
597   \let\urlName\rma@urlresource
598   \makeJSSpecials
```

Empty these to macros for they might contain content from a previous annot with resources.

```
599   \let\rma@addResources\@empty\let\rma@addFileSpecs\@empty
```

We begin by passing the dimensions through a length so the author can use the `calc` package.

```
600   \bgroup
601     \setlength{\dimen@}{#2}\xdef\rm@Annot@width{\the\dimen@}%
602     \setlength{\dimen@}{#3}\xdef\rm@Annot@height{\the\dimen@}%
603   \egroup
```

Next we increment a running counter, to give each annot, and all indirect references a unique name.

```
604   \stepcounter{rm@Cnt}%
```

We take the 4th parameter, a named path, and pass it to `\useNamedPath`, then use `\filename@parse` to extract the components of the path.

```
605   \rma@edefexecute{\noexpand\filename@parse{\rma@useNamedPath{#4}}}%
```

We record this named path as `rmFileStrm#4`, if this command is undefined, we define it. It's value is the indirect reference to the stream; if already defined, the command should contain the indirect reference to the same file that has been already embedded. If undefined we set a boolean to embed this stream, otherwise, we don't embed.

```
606   \edef\rma@fs@expand{rmFileStrm#4}\ifundefined{\rma@fs@expand}{%
607   \rm@csarg\xdef{\rma@fs@expand}%
608     {rmfstream\therm@Cnt}\global\rma@EmbedFiletrue}%
609     {\global\rma@EmbedFilefalse}%
```

After having parsed the path, we now save the pieces for later use.

```

610 \edef\rma@thisfilepath{\filename@area}%
611 \edef\rma@basefilename{\filename@base}%
612 \edef\rma@extension{\filename@ext}%
613 \rma@edefexecute{\noexpand\uppercase{\noexpand
614   \def\noexpand\rma@tempi{\rma@extension}}}%

```

Take a look at the file extension, if it is a 3D type extension, mark it as a RM3D annot by setting `\rma@isiiidtrue`.

```

615 \ifx\rma@tempi\rma@rmAnnot@type@uiiid\rma@isiiidtrue
616   \else\ifx\rma@tempi\rma@rmAnnot@type@prc\rma@isiiidtrue
617   \else\rma@isiiidfalse
618 \fi\fi

```

Process Options. We finally get around to processing the options. We put `\setkeys` in an `\edef` to allow the user to use macros to specify some of the options. The next line are the options passed by #1 of `\rmAnnot`.

```

619 \rma@edefexecute{\noexpand\setkeys{rmAnnot}{#1}}%

```

If use has specified either the `width` or `height`, we reset the width and height of the annotation, while preserving the aspect ratio. We first test `width` then `height`. If both are specified, we use the `width` key and ignore the `height` key.

```

620 \bgroup
621 \ifx\rmAnnot@width\@empty
622   \ifx\rmAnnot@height\@empty

```

If `width` and `height` are not specified, we determine if the `scale` key is listed.

```

623   \ifx\rmAnnot@scale\@empty\else
624     \setlength{\dimen@}%
625       {\rmAnnot@height*\real{\rmAnnot@scale}}%
626     \xdef\rmAnnot@height{\the\dimen@}%
627     \setlength{\dimen@}%
628       {\rmAnnot@width*\real{\rmAnnot@scale}}%
629     \xdef\rmAnnot@width{\the\dimen@}%
630   \fi
631 \else

```

If `height` but not `width` is specified, we resize accordingly.

```

632   \setlength{\dimen@}%
633     {\rmAnnot@height*\ratio
634     {\rmAnnot@width}{\rmAnnot@height}}%
635   \xdef\rmAnnot@width{\the\dimen@}%
636   \setlength{\dimen@}{\rmAnnot@height}%
637   \xdef\rmAnnot@height{\the\dimen@}%
638 \fi
639 \else

```

If `width` but not `height` is specified, we resize accordingly.

```

640   \setlength{\dimen@}%
641     {\rmAnnot@width*\ratio
642     {\rmAnnot@height}{\rmAnnot@width}}%

```

```

643 \xdef\rma@Annot@height{\the\dimen@}%
644 \setlength{\dimen@}{\rmAnnot@width}%
645 \xdef\rma@Annot@width{\the\dimen@}%
646 \fi
647 \egroup
648 \ifx\rma@input@iiidCode\relax\else
649 \ifrma@isiiid

```

Process the options of the RM3D annot, as passed to us by `\setRmOptions3D`.

```

650 \@ifundefined{\rma@Annot@name_3DOPTS}{%

```

To Do. Here, we can insert some default options if the user did not specify anything. Including a reminder to make a declaration.

```

651 }{%

```

If the user has specified the `use3D` option and the file specified in #4 is a 3D model (U3D or PRC), then the switch `\ifrma@isiiid` is true, and we process any 3D options as specified by `\setRmOptions3D`. The command sequence `\rma@Annot@name_3DOPTS` is defined for this annot by `\setRmOptions3D`; we only process if there are options for this annot.

```

652 \def\rma@Instances{}\def\rma@appendToNameTree{%

```

Expand the arguments of `\setkeys` before allowing `\setkeys` to execute.

```

653 \rma@edefexecute{\noexpand\setkeys{rm3DOptsTopLevel}%
654 {\@nameuse{\rma@Annot@name_3DOPTS}}}%
655 \edef\additional@Instances{\rma@Instances}%
656 \edef\rma@addResources{\rma@appendToNameTree}%

```

3Djsript: We determine if there is one or more javascript files specified.

```

657 \ifx\rma@rmAnnot@iiidJs\@empty\else
658 \def\@MXV@jscriptiiid{%
659 \literalps@out{%
660 \ps@mark/_objdef {jscriptiiid\therm@Cnt}%
661 /type/array/OBJ pdfmark^^J%
662 \rmiid@addToScriptsArray
663 \rma@addFileSpecs
664 }%

```

We save the key-value pair for the `Scripts` key.

```

665 \edef\@MXV@jscriptiiid{%
666 /Scripts {jscriptiiid\therm@Cnt}%
667 }%
668 \fi

```

Build the array of 3D views

```

669 \@MXV@buildva%
670 }%
671 \fi\fi

```

Identify the Extension. We try to identify the extension provided by the author. We take the `\filename@base` (saved as `\rma@extension`), convert it to upper case and compare with the text macros containing SWF, FLV, MP3, or one of the other

supported extensions. The macro `\rma@tempi` contains the upper case form of the extension.

```
672 \rma@edefexecute{\noexpand\uppercase{\noexpand
673 \def\noexpand\rma@tempi{\rma@extension}}}%
```

3D Model. Search for a U3D or PRC. We earlier did a test to see if this is 3D or not, so the boolean `\ifrma@isiid` has already been set.

```
674 \ifrma@isiid
675 \def\rma@poster@descrip{3D}%
676 \def\rma@RMCSsubtype{/3D}%
677 \ifx\rma@tempi\rma@rmAnnot@type@uiid
678 \def\rma@rmAnnot@type{U3D}%
679 \edef\rma@mimeType{\rma@mimetype@uiid}%
680 \else
681 \ifx\rma@tempi\rma@rmAnnot@type@prc
682 \def\rma@rmAnnot@type{PRC}%
683 \edef\rma@mimeType{\rma@mimetype@prc}%
684 \fi\fi
685 \else
```

Not a 3D model, so we'll check for more conventional types.

Flash Application. Search for a SWF file, we set the identifiers for later use.

```
686 \def\rma@poster@descrip{Flash}%
687 \ifx\rma@tempi\rma@rmAnnot@type@swf
688 \def\rma@rmAnnot@type{SWF}\edef\rma@mimeType{\rma@mimetype@swf}%
689 \def\rma@RMCSsubtype{/Flash}%
690 \else % if not flash
```

Video Formats. We search for extension that is generally classified as video.

```
691 \def\rma@poster@descrip{Video}\def\rma@rmAnnot@type{FLV}%
692 \@FndSuppExtfalse
693 \@tfor\rma@type:={flv}{fiv}{mpiv}{mivV}{mov}{iiiGP}{iiiGii}\do{%
694 \expandafter\ifx\expandafter\rma@tempi\csname%
695 rma@rmAnnot@type@\rma@type\endcsname
696 \@FndSuppExttrue
697 \edef\rma@mimeType{\csname%
698 rma@mimetype@\rma@type\endcsname}%
699 \rma@recordVideoPlayer\@break@tfor
700 \fi
701 }%
702 \if@FndSuppExt
703 \def\rma@RMCSsubtype{/Video}%
704 \xdef\FileStrmVideoPlayer{rmVideoPlayer\therm@Cnt}%
705 \else % if not video
```

Audio Formats. Test for a MP3 file, we need to embed the `AudioPlayer` once and only once. The Boolean `\ifrma@EmbedAudioPlayer` and the marker `\rma@isAPEEmbedded` are used to keep track of whether the player has been embedded.

```

706 \ifx\rma@tempi\rma@rmAnnot@type@mpiii
707   \def\rma@rmAnnot@type{MP3}\edef\rma@mimeType{\rma@mimetype@mpiii}%
708   \def\rma@poster@descrip{MP3}\def\rma@RMCSsubtype{/Sound}%
709   \xdef\FileStrmAudioPlayer{rmAudioPlayer\therm@Cnt}%
710   \let\rma@rmAnnot@resources\@empty
711   \rma@recordAudioPlayer
712 \else % not mp3

```

Error. The extension is not recognized.

```

713   \rma@PkEr@i
end testing for 3D (u3d and prc), flash, video, and audio
714 \fi\fi\fi\fi

```

Define `\rma@thisfileName` and `\rma@fullpath` for later use.

```

715 \def\rma@thisfileName{\rma@basefilename.\rma@extension}%
716 \def\rma@fullpath{\rma@thisfilepath\rma@thisfileName}%

```

If this is an FLV video file, we don't let the user created flash variables

```

717 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
718   \let\rma@rmAnnot@flashvars\@empty\fi
719 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@mpiii
720   \let\rma@rmAnnot@flashvars\@empty\fi

```

If there is no poster, we supply one. For MP3, it is an image of the `AudioPlayer` controls, otherwise, it is the default poster.

```

721 \ifx\rma@rmAnnot@poster\@empty
722   \ifKV@rmAnnot@defaultposter
723     \Gin@defaultbp\this@width\rm@Annot@width
724     \Gin@defaultbp\this@height\rm@Annot@height
725     \ifdim\rm@Annot@width < \rm@Annot@height
726       \edef\calc@prop{\this@width}\else
727       \edef\calc@prop{\this@height}\fi
728     \def\this@bbox{0 0 \this@width\space\this@height}%
729     \begin{sp@createImage}{\this@bbox}{rmAP@#4@\therm@Cnt}%
730     \rma@invisible
731     \rma@psgraphics@poster
732     \end{sp@createImage}%
733     \def\rma@rmAnnot@poster{rmAP@#4@\therm@Cnt}%
734 \else
735   \ifx\rma@rmAnnot@type\rma@rmAnnot@type@mpiii
736     \def\rma@rmAnnot@poster{nramp3poster}%
737   \else
738     \Gin@defaultbp\this@width\rm@Annot@width
739     \Gin@defaultbp\this@height\rm@Annot@height
740     \ifdim\rm@Annot@width < \rm@Annot@height

```

```

741         \edef\calc@prop{\this@width}\else
742         \edef\calc@prop{\this@height}\fi
743         \def\this@bbox{0 0 \this@width\space\this@height}%
744         \begin{sp@createImage}{\this@bbox}{rmAP@#4@\therm@Cnt}%
745         \rma@invisible
746         \rma@psgraphics@poster
747         \end{sp@createImage}%
748         \def\rma@rmAnnot@poster{rmAP@#4@\therm@Cnt}%
749     \fi
750 \fi
751 \fi

```

Begin the construction of the RMA. Place the dimensions input by the author in a `\Bbox` within `\pdf@rect`. `\Bbox` is defined in the `eforms` package.

```

752 \pdf@rect{\Bbox{\rm@Annot@width}{\rm@Annot@height}}%
753 \@MXV@newlabel{rmAnnot_\rma@Annot@name}{rmAnnot\therm@Cnt}%
754 \@MXV@labeltoaux{rmAnnot_\rma@Annot@name}{rmAnnot\therm@Cnt}%

```

Begin writing the rich media annotation through a PostScript special. The command `\literalps@out` is defined in `hyperref`.

```

755 \literalps@out{%

```

Create the RichMedia Annotation.

```

756 \ps@mark/_objdef {rmAnnot\therm@Cnt}%
757 /Type/Annot%
758 /Subtype/RichMedia%
759 /NM (\rma@Annot@name)%           % Annotation name
760 \ifx\rma@rmAnnot@poster\@empty\else
761 /AP <</N {\rma@rmAnnot@poster}>>% % poster appearance
762 \fi
763 /F 68%                           % Annotation flags
764 /P {ThisPage}%                   % Parent
765 /Border [ 0 0 \rma@rmAnnot@borderwidth ]% Border
766 /BS <</Type/Border%             % Border Style dictionary
767 /W \rma@rmAnnot@borderwidth%     % Width
768 /S/S%                           % Border style (Solid)
769 >>%

```

The **RichMedia** annot has a **RichMediaContent** dictionary, and a **RichMediaSettings** dictionary, give indirect references to these.

```

770 /RichMediaContent {rmContent\therm@Cnt}%
771 /RichMediaSettings {rmSettings\therm@Cnt}%
772 H.B /ANN pdfmark^^J%

```

The RichMediaContent dictionary. We include the **Configurations** key, an array of indirect references to **RichMediaConfiguration** dictionaries, and the **Assets** key, an indirect reference to the **Assets** dictionary.

```

773 \ps@mark/_objdef {rmContent\therm@Cnt}/type/dict/OBJ pdfmark^^J%
774 \ps@mark{rmContent\therm@Cnt} <<%
775 /Type/RichMediaContent%

```

If this is a RM3DA, we declare it in the `RichMediaContent` dictionary. Here we declare `/Subtype/3D` and insert a views array.

```

776 \ifirma@isiiid
777     /Subtype/3D%
778 \ifx\@MXV@varray\@empty\else
779     /Views [\@MXV@varray]%
780 \fi\fi
781     /Configurations [{rmConfig\therm@Cnt}]%
782     /Assets {rmAssets\therm@Cnt}%
783 >>/PUT pdfmark^^J%

```

The RichMediaConfiguration dictionary. We set the primary content type of the configuration (Flash, Video, or Sound), and reference the corresponding instances, the SWF file to play, the VideoPlayer to play a FLV file, and the AudioPlayer, to play MP3 files.

```

784     \ps@mark/_objdef {rmConfig\therm@Cnt}/type/dict/OBJ pdfmark^^J%
785     \ps@mark{rmConfig\therm@Cnt} <<%
786         /Type/RichMediaConfiguration%
787         /Name (RMConfig\therm@Cnt)%

```

The Subtype is 3D, Flash, Video, or Sound, here, `\rma@RMCSUBTYPE` was determined earlier.

```

788         /Subtype\rma@RMCSUBTYPE%
789         /Instances {rmInstances\therm@Cnt}%
790 >> /PUT pdfmark^^J%

```

The Instances Array.

This version of the Instances array is used, same as the SWF version.

```

791     \ps@mark/_objdef {rmInstances\therm@Cnt}/type/array/OBJ pdfmark^^J%
792 \ifirma@isiiid
793     \ps@mark{rmInstances\therm@Cnt} {rmInstance\therm@Cnt}%
794     /APPEND pdfmark^^J%
795     \additional@Instances
796 \else

```

We load the indirect reference to the VideoPlayer

```

797 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
798     \ps@mark{rmInstances\therm@Cnt} {rmVideoPlayer\therm@Cnt}%
799     /APPEND pdfmark^^J%
800 \else\ifx\rma@rmAnnot@type\rma@rmAnnot@type@swf

```

The Instances array will be populated later by resources.

```

801     \ps@mark{rmInstances\therm@Cnt} {rmInstance\therm@Cnt}%
802     /APPEND pdfmark^^J%
803 \else

```

We load the indirect reference to the AudioPlayer

```

804     \ps@mark{rmInstances\therm@Cnt} {rmAudioPlayer\therm@Cnt}%
805     /APPEND pdfmark^^J%
806 \fi\fi\fi

```

The Assets name tree. Reference by the RichMediaContent dictionary, this name tree lists the files needed, and indirect references to their respective file specification dictionaries. The assets are a function of whether the file type is SWF, FLV or MP3. For FLV, there are a number of skins that can be selected from, so the skin select through the skin key is included as assets, and its corresponding SWF needs to be embedded, if not embedded already.

```

807 \ps@mark/_objdef {rmAssets\therm@Cnt}/type/dict/OBJ pdfmark^^J%
808 \ps@mark{rmAssets\therm@Cnt} <<%
809 /Names {\rma@ANT}>>/PUT pdfmark^^J%
810 \ps@mark/_objdef {\rma@ANT}/type/array/OBJ pdfmark^^J%
811 \ifrma@isiiid
812 \ps@mark{\rma@ANT} (\rma@thisfileName) /APPEND pdfmark^^J%
813 \ps@mark{\rma@ANT} {rmfilespec\therm@Cnt} /APPEND pdfmark^^J%
814 \rma@addResources
815 \else
816 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@swf
817 \ifKV@rmAnnot@url\else
818 \ps@mark{\rma@ANT} (\rma@thisfileName)/APPEND pdfmark^^J%
819 \ps@mark{\rma@ANT} {rmfilespec\therm@Cnt}/APPEND pdfmark^^J%
820 \rma@addResources
821 \fi
822 \else\ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
823 \ifKV@rmAnnot@url\else
824 \ps@mark{\rma@ANT} (\rma@thisfileName)/APPEND pdfmark^^J%
825 \ps@mark{\rma@ANT} {rmfilespec\therm@Cnt}/APPEND pdfmark^^J%
826 \ifVideoPlayerEx\rma@addResources\fi
827 \fi
828 \ifx\rma@rmAnnot@Skin\@empty\else
829 \ps@mark{\rma@ANT} (\rma@rmAnnot@Skin)/APPEND pdfmark^^J%
830 \ps@mark{\rma@ANT} {rmfilespecSkin\rma@skinName}%
831 /APPEND pdfmark^^J%
832 \fi
833 \ps@mark{\rma@ANT} (\rma@VideoPlayer)/APPEND pdfmark^^J%
834 \ps@mark{\rma@ANT} {rmfilespecVP}/APPEND pdfmark^^J%
835 \else
836 \ps@mark{\rma@ANT} (AudioPlayer.swf)/APPEND pdfmark^^J%
837 \ps@mark{\rma@ANT} {rmfilespecAP}/APPEND pdfmark^^J%
838 \ps@mark{\rma@ANT} (\rma@thisfileName)/APPEND pdfmark^^J%
839 \ps@mark{\rma@ANT} {rmfilespec\therm@Cnt}/APPEND pdfmark^^J%
840 \fi\fi\fi

```

The RichMediaInstance dictionary. Describes a single instance of an asset. The asset being described is the VideoPlayer (in the case of FLV files), the Flash file in the case of SWF files, and the AudioPlayer (for MP3 files).

```

841 \ifrma@isiiid
842 \ps@mark/_objdef {rmInstance\therm@Cnt}/type/dict/OBJ pdfmark^^J%
843 \ps@mark{rmInstance\therm@Cnt}%
844 \else
845 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv

```

```

846 \ps@mark/_objdef {rmVideoPlayer\therm@Cnt}/type/dict/OBJ pdfmark^^J%
847 \ps@mark{rmVideoPlayer\therm@Cnt}%
848 \else\ifx\rma@rmAnnot@type\rma@rmAnnot@type@swf
849 \ps@mark/_objdef {rmInstance\therm@Cnt}/type/dict/OBJ pdfmark^^J%
850 \ps@mark{rmInstance\therm@Cnt}%
851 \else
852 \ps@mark/_objdef {rmAudioPlayer\therm@Cnt}/type/dict/OBJ pdfmark^^J%
853 \ps@mark{rmAudioPlayer\therm@Cnt}%
854 \fi\fi\fi
855 <<%
856 /Type/RichMediaInstance%
857 \ifrma@isiiid
858 /Subtype/3D%
859 /Asset {rmfilespec\therm@Cnt}%
860 \else
861 /Subtype/Flash%
862 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
863 /Asset {rmfilespecVP}%
864 \else\ifx\rma@rmAnnot@type\rma@rmAnnot@type@swf
865 /Asset {rmfilespec\therm@Cnt}%
866 \else
867 /Asset {rmfilespecAP}%
868 \fi\fi\fi
869 \ifrma@isiiid\else
870 /Params {rmParams\therm@Cnt}%
871 \fi
872 >> /PUT pdfmark^^J%

```

The Params dictionary. Contains parameters related to an active Flash subtype in a RichMediaInstance dictionary; the user, through the UI or through this package, can set the value of the FlashVars entry. For content subtypes other than Flash, the argument for FlashVars is not under user control, and is determined by the PDF creation application, this package.

```

873 \ps@mark/_objdef {rmParams\therm@Cnt}/type/dict/OBJ pdfmark^^J%
874 \ps@mark{rmParams\therm@Cnt} <<%
875 /Type/RichMediaParams%

```

If there are no additional resources specified, we bind the animation to the background, if there are (SWF) resources, we bind to the foreground.

```

876 \ifrma@isiiid\else
877 \ifx\rma@rmAnnot@resources\@empty
878 /Binding/Background%
879 \else
880 /Binding/Foreground%
881 \fi\fi

```

If this is a FLV (video), we use the custom flash variables of Acrobat (reverse engineering). If it is a url, we specify the full URL, otherwise, we specify the file name as the source.

```

882 \ifrma@isiiid\else

```

```

883 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
884 \ifKV@rmAnnot@url
885 /FlashVars (source=\rma@fullpath&%
886 \else
887 /FlashVars (source=\rma@thisfileName&%
888 \fi
889 \ifx\rma@rmAnnot@Skin\@empty\else
890 skin=\rma@rmAnnot@Skin&%
891 \fi
892 skinAutoHide=\rma@skinAutoHide&%
893 skinBackgroundColor=\rma@skinBGColor&%
894 skinBackgroundAlpha=\rma@skinBGAlpha&%
895 volume=\rma@rmAnnot@volume)
896 \ifx\rma@rmAnnot@cuepoints\@empty\else
897 /CuePoints [\rma@array@hold]%
898 \fi
899 \else\ifx\rma@rmAnnot@type\rma@rmAnnot@type@swf

```

If this is a SWF file, we allow the author to introduce custom flash variables, hope s/he knows what is s/he is doing.

```

900 \ifx\rma@rmAnnot@flashvars\@empty\else
901 /FlashVars (\rma@rmAnnot@flashvars)%
902 \fi
903 \else
904 /FlashVars (source=\ifKV@rmAnnot@url\rma@fullpath\else
905 \rma@thisfileName\fi&autoPlay=true&%
906 volume=\rma@rmAnnot@volume)%
907 \fi\fi\fi
908 >> /PUT pdfmark^^J%

```

The RichMediaSettings Dictionary. The second dictionary referenced in the RichMedia annot is the RichMediaSettings dictionary. The RichMediaSettings dictionary stores the conditions and responses that occur in response to certain events, such as activation and deactivation of the annotation, and contains two dictionaries.

```

909 \ps@mark/_objdef {rmSettings\therm@Cnt}/type/dict/OBJ pdfmark^^J%
910 \ps@mark{rmSettings\therm@Cnt} <<%
911 /Type/RichMediaSettings%

```

The Activation key is a dictionary that describes how the annot is to be activated and played.

```

912 /Activation <<%
913 /Type/RichMediaActivation%
914 /Condition\rma@rmAnnot@enabled
915 /Configuration {rmConfig\therm@Cnt}%

```

(2011/11/08) Used for Keyframe animation, normally not needed. Will uncom-
ment when I develop an example of usage. Note, the speed and playcount keys
are now ignored.

```

916 \ifrma@isiiid

```

```

917         /Animation%
918         <<%
919         /Type/RichMediaAnimation%
920         /Subtype/Linear%
921         /Speed \rma@rmAnnot@speed
922         /PlayCount \rma@rmAnnot@playcount
923         >>%
924 \ifx\@MXV@defaultview\@empty\else
925     /View \@MXV@defaultview
926 \fi
927 \ifx\@MXV@jscriptiid\@empty\else
928     \@MXV@jscriptiid
929 \fi\fi
930 /Presentation {rmPresentation\therm@Cnt}%
931 >>

```

The Deactivation key is a dictionary that describes how the annot is to be deactivated.

```

932     /Deactivation<<%
933     /Type/RichMediaDeactivation%
934     /Condition\rma@rmAnnot@deactivated
935     >>%
936 >>/PUT pdfmark^^J%

```

The CuePoints Array.

```

937 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
938 \ifx\rma@rmAnnot@cuepoints\@empty\else\rma@dict@hold\fi\fi

```

The RichMediaPresentation Dictionary. Referenced within the Activation entry of the RichMediaSettings dictionary above, this dictionary determines whether the background is transparent, and whether the media is to be played within the page or in a floating window.

```

939 \ps@mark/_objdef {rmPresentation\therm@Cnt}%
940 /type/dict/OBJ pdfmark^^J%
941 \ps@mark{rmPresentation\therm@Cnt}<<%
942 /Type/RichMediaPresentation%
943 \ifrma@isiiid
944 /NavigationPane \rma@rmAnnot@modeltree % need key
945 /Toolbar \rma@rmAnnot@toolbar % need key
946 /Transparent \rma@rmAnnot@transparent
947 \else
948 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@swf
949     /Transparent \rma@rmAnnot@transparent
950     /PassContextClick \rma@rmAnnot@PassContextClick
951     /NavigationPan false%
952 \else
953     /Transparent false%
954     /NavigationPan false%
955 \fi\fi
956 \ifKV@rmAnnot@windowed

```

```

957     /Style/Windowed%
958     /Window {rmWindow\therm@Cnt}%
959 \else
960     /Style/Embedded%
961 \fi
962 >>/PUT pdfmark^^J%

```

The RichMediaWindow Dictionary. When the style is Windowed as specified in the RichMediaPresentation dictionary, we include the RichMediaWindow dictionary to set the parameters of the window. Currently, the parameters are hard-wired to the defaults, as specified by the *Adobe Suppl. Doc*.

```

963 \ifKV@rmAnnot@windowed
964   \ps@mark/_objdef {rmWindow\therm@Cnt}/type/dict/OBJ pdfmark^^J%
965   \ps@mark{rmWindow\therm@Cnt}<<%
966     /Type/RichMediaWindow%
967     /Height<<%
968       /Default \rma@winDimPosHeight@def
969       /Max \rma@winDimPosHeight@max
970       /Min \rma@winDimPosHeight@min
971     >>%
972     /Width<<%
973       /Default \rma@winDimPosWidth@def
974       /Max \rma@winDimPosWidth@max
975       /Min \rma@winDimPosWidth@min
976     >>%
977     /Position<<%
978       /Type/RichMediaPosition % RichMediaPosition dictionary
979       /HAlign\rma@winDimPosPos@halign
980       /VAlign\rma@winDimPosPos@valign
981       /HOffset \rma@winDimPosPos@hoffset
982       /VOffset \rma@winDimPosPos@voffset
983     >>%
984     >>/PUT pdfmark^^J%
985 \fi

```

File Specifications and Streams. In this section, we insert the Filespec dictionaries of the various assets. There are four type of assets: (1) the one specified by the 4th argument of the \rmAnnot; (2) additional assets specified through the **resources** key; (3) the VideoPlayer; and (4) the AudioPlayer. The first two we'll call *Author Supplied Assets*, the latter two, we'll call *System Supplied Assets*.

Author Supplied Assets. We first insert the file spec and stream for the 4th parameter, its filename is \rma@thisfileName and its full path name is \rma@fullpath.

File specs: Filespec dictionary

```

986   \ps@mark/_objdef {rmfilespec\therm@Cnt}/type/dict/OBJ pdfmark^^J%
987   \ps@mark{rmfilespec\therm@Cnt} <<%
988   \ifKV@rmAnnot@url

```

```

989     /F(\rma@fullpath)%
990     /FS/URL%
991 \else
992     /F(\rma@thisfileName)%
993     /UF (\rma@thisfileName)%
994     /EF <</F {\csname rmFileStrm#4\endcsname}>>
995 \fi
996     /Type/Filespec
997 >>/PUT pdfmark^^J%

```

File stream: EmbeddedFile dictionary

```

998 \ifKV@rmAnnot@url\else
999 \ifrma@EmbedFile
1000 \ps@mark/_objdef {\csname rmFileStrm#4\endcsname}%
1001     /type/stream/OBJ pdfmark^^J%
1002 \ps@mark{\csname rmFileStrm#4\endcsname} (\rma@fullpath)
1003     (r) file /PUT pdfmark^^J%
1004 \ps@mark{\csname rmFileStrm#4\endcsname} <<%
1005     /Type/EmbeddedFile%
1006     /Subtype(\rma@mimeType)%
1007 >>/PUT pdfmark^^J%
1008 \ps@mark{\csname rmFileStrm#4\endcsname} /CLOSE pdfmark^^J%
1009 \fi\fi

```

If we are dealing with a SWF or 3D file, we'll then include additional file specs and streams as specified by the `resources` key of `\rmAnnot`.

```

1010 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@swf\rma@addFileSpecs\fi
1011 \ifrma@isiiid\rma@addFileSpecs\fi

```

System Supplied Assets. We have the various skins, the `VideoPlayer`, and the `AudioPlayer`. We'll start with the skins.

File specs for skin

```

1012 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
1013 \ifVideoPlayerEx\rma@addFileSpecs\fi
1014 \ifx\rma@rmAnnot@Skin\@empty\else
1015 \ps@mark/_objdef {rmfilespecSkin\rma@skinName}%
1016     /type/dict/OBJ pdfmark^^J%
1017 \ps@mark{rmfilespecSkin\rma@skinName} <<%
1018     /F (\rma@rmAnnot@Skin)%
1019     /Type/Filespec%
1020     /UF (\rma@rmAnnot@Skin)%
1021     /EF <</F {rmfstreamSkin\rma@skinName}>>
1022 >>/PUT pdfmark^^J%

```

File stream for skin

```

1023 \rm@csarg\if{embedSkin\rma@skinName}\rm@One
1024 \ps@mark/_objdef {rmfstreamSkin\rma@skinName}%
1025     /type/stream/OBJ pdfmark^^J%
1026 \ps@mark{rmfstreamSkin\rma@skinName}%
1027     (\rma@pathToSkins/\rma@rmAnnot@Skin) (r) file%

```

```

1028     /PUT pdfmark^^J%
1029     \ps@mark{rmfstreamSkin\rma@skinName} <<%
1030     /Type/EmbeddedFile
1031     /Subtype (\rma@mimetype@swf)
1032     >>/PUT pdfmark^^J%
1033     \ps@mark{rmfstreamSkin\rma@skinName}/CLOSE pdfmark^^J%
1034 \fi\fi\fi

```

Now the specs and stream of the VideoPlayer

File specs for video player

```

1035 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@flv
1036     \ps@mark/_objdef {rmfilespecVP}/type/dict/OBJ pdfmark^^J%
1037     \ps@mark{rmfilespecVP} <<%
1038     /Type/Filespec%
1039     /F (\rma@VideoPlayer)%
1040     /UF (\rma@VideoPlayer)%
1041     /EF <</F {rmfstreamVP}>>%
1042     >>/PUT pdfmark^^J%

```

We'll only embed once, provided \ifrma@EmbedVideoPlayer is true.

File stream for video player

```

1043 \ifrma@EmbedVideoPlayer
1044     \ps@mark/_objdef {rmfstreamVP}/type/stream/OBJ pdfmark^^J%
1045     \ps@mark{rmfstreamVP} (\rma@pathToPlayers\rma@VideoPlayer)
1046     (r) file /PUT pdfmark^^J%
1047     \ps@mark{rmfstreamVP} <<%
1048     /Type/EmbeddedFile%
1049     /Subtype (\rma@mimetype@swf)%
1050     >>/PUT pdfmark^^J%
1051     \ps@mark{rmfstreamVP} /CLOSE pdfmark^^J%
1052 \fi\fi

```

Now the specs and stream of the AudioPlayer

File specs for audio player

```

1053 \ifx\rma@rmAnnot@type\rma@rmAnnot@type@mpiii
1054     \ps@mark/_objdef {rmfilespecAP}/type/dict/OBJ pdfmark^^J%
1055     \ps@mark{rmfilespecAP} <<%
1056     /F (AudioPlayer.swf)%
1057     /Type/Filespec%
1058     /UF (AudioPlayer.swf)%
1059     /EF <</F {rmfstreamAP}>>%
1060     >>/PUT pdfmark^^J%

```

We'll only embed once, provided \ifrma@EmbedAudioPlayer is true.

File stream for audio player

```

1061 \ifrma@EmbedAudioPlayer
1062     \ps@mark/_objdef {rmfstreamAP}/type/stream/OBJ pdfmark^^J%
1063     \ps@mark{rmfstreamAP}(\rma@pathToPlayers/AudioPlayer.swf)
1064     (r) file /PUT pdfmark^^J%

```

```

1065 \ps@mark{rmfstreamAP} <<%
1066     /Type/EmbeddedFile%
1067     /Subtype (\rma@mimetype@mpiii)%
1068     >>/PUT pdfmark^^J%
1069 \ps@mark{rmfstreamAP} /CLOSE pdfmark^^J%
1070 \fi\fi}%

```

If we are using user defined skins, we set the skin just used so that this skin will not be embedded a second time, let's hope.

```

1071 \ifx\rma@rmAnnot@type@flv\rma@rmAnnot@type
1072 \ifx\rma@rmAnnot@Skin\@empty\else
1073 \expandafter\global\rma@csarg\let{embedSkin\rma@skinName}\rm@Zero
1074 \fi\fi

```

End of \rmAnnot. Close off the group, and end the \rmAnnot command definition.

```

1075 \endgroup}

```

4.3 Poster Appearances

\defaultPoster The command `\defaultPoster` defines the default poster appearance and is used when a poster appearance is not specified by the `poster` key. If the file is an MP3, then there is a graphic that is the default poster appearance.

`\defaultPoster` can be defined to something else, if desired. The commands should be PostScript graphic operators.

This command defines a text macro `\rma@psgraphics@poster` that is expanded in the definition of `\rmAnnot`, before the start of the `annot` itself.

```

1086 \newcommand{\defaultPoster}[1]{\def\rma@psgraphics@poster{#1}}
1087 \defaultPoster
1088 {%
1089     \rma@ps@bg@setcolor
1090     0 0 \this@width\space\this@height\space rectfill
1091     \rma@ps@txt@x\adj@measure\rma@ps@txt@y\adj@measure moveto
1092     \rma@ps@txt@setcolor/\rma@ps@font
1093     \rma@ps@relfontsize\rma@ps@fontsize selectfont
1094     \rma@ps@msg
1095 }

```

The definitions of the text macros that enable the document author to make minor changes in color, font, and placement of text, to the default poster.

```

1086 \def\adj@measure{\calc@prop\space mul 100 div }
1087 \def\rma@ps@bg@setcolor{.7529 setgray }
1088 \def\rma@ps@txt@x{10 }\def\rma@ps@txt@y{10 }
1089 \def\rma@ps@txt@setcolor{.4 setgray }
1090 \def\rma@ps@font{Helvetica }
1091 \def\rma@ps@relfontsize{10 \adj@measure}
1092 \let\rma@ps@fontsize\@empty
1093 \def\rma@ps@msg{(\rma@posternote) show}

```

\setPosterProps{<KV-pairs>} A convenience command to execute `<KV-pairs>` from the `rmPoster`

family. Below are the key-values of the `rmPoster` family for designing your own custom default poster. All values are expressed in PostScript operators.

```

1094 \define@key{rmPoster}{color}[]{\def\rma@ps@bg@setcolor{#1 }}
1095 \define@key{rmPoster}{xPos}[]{\def\rma@ps@txt@x{#1 }}
1096 \define@key{rmPoster}{yPos}[]{\def\rma@ps@txt@y{#1 }}
1097 \define@key{rmPoster}{textColor}[]{\def\rma@ps@txt@setcolor{#1 }}
1098 \define@key{rmPoster}{relTextSize}[]{\def
1099     \rma@ps@relfontsize{#1 \adj@measure}%
1100     \let\rma@ps@fontsize\empty}
1101 \define@key{rmPoster}{textSize}[]{\def\rma@ps@fontsize{#1 }}
1102     \let\rma@ps@relfontsize\empty}
1103 \define@key{rmPoster}{textFont}[]{\def\rma@ps@font{#1 }}
1104 \def\setPosterProps#1{\setkeys{rmPoster}{#1}}
```

`\makePoster` A convenience command for making posters. Assuming you have an eps of the appropriate aspect ratio to use as poster, you can say

```
\makePoster[<graphics_options>]{<name>}{<graphics_file>}
```

for example

```
\makePoster[hiresbb]{AcroAd_poster}{AcroAd_poster}
```

Then you can say

```
\rmAnnot[poster=AcroAd_poster]{612bp}{265bp}{AcroAd}
```

```

1105 \providecommand{\makePoster}[3][]{%
1106     \embedEPS[#1]{rma@#2}{#3}%
1107     \begin{createImage}{\bbox0f{rma@#2}{#2}%
1108         \ps@mark{rma@#2} /SP pdfmark
1109     \end{createImage}%
1110 }
1111 \@onlypreamble{\makePoster}
```

Finally, we define several error messages.

```

1112 \def\rma@PkEr@i{%
1113     \PackageError{rmannot}{%
1114         You must specify a file with an extension\MessageBreak
1115         of .swf, .flv, .f4v, .mp4, .m4v, .mov, .3gp,\MessageBreak
1116         .3g2, .mp3}{Specify one of the supported file extensions to
1117         embed in this annotation.\MessageBreak
1118         See the rmannot manual for details on supported extensions.}}
1119 \def\rma@PkEr@ii{%
1120     \PackageError{rmannot}{%
1121         The name '\rm@cargii' has already been used. Either\MessageBreak
1122         you are defining the same path, or a different path\MessageBreak
1123         with the same name}{%
1124         Names must be unique to the document, choose another}}
1125 \def\rma@PkEr@iii#1{%
1126     \PackageError{rmannot}{%

```

```

1127     No extension supplied with this file name,\MessageBreak#1.%
1128     \MessageBreak Please include a file extension of\MessageBreak
1129     .swf, .flv, or .mp3, as appropriate}{%
1130     Include an extension of .swf, .flv, or .mp3}}
1131 </package>

```

4.4 Support for 3D Annotations

This section supports options of the command `\setRmOptions3D`, used for passing a set of options to a RM3DA. At this writing, there are two recognized keys, `3DResources` and `3DOptions`. The former holds the key-values of my own construction; the latter holds the key-values of the `movie15` package by Alexander Grahn.

```

1132 <*3Dcode>

\setRmOptions3D{myDice}
{
    3DOptions={\options-from-movie15},
    3DResources={
        none={rName=<name1>} ,... ,
        foreground={rName=<name2>,flashvars=<vars>} ,... ,
        background={rName=<name3>,flashvars=<vars>} ,... ,
        material={rName=<name4>,mName=<materialName>,flashvars=<vars>} ,...
    }
}

```

`3DOptions` (3D) The `rm3DOptsTopLevel` family supports to top-level keys of `\setRmOptions3D`:
`3DResources` (3D) `3DOptions` and `3DResources`.

```

1133 \define@key{rm3DOptsTopLevel}{3DOptions}{%
1134 \def\rmiidTLOpts{#1}%

```

This `\define@key`, in turn, executes the family `MXV@user`, taken from `movie15`.

```

1135 \setkeys{MXV@user}{#1}%
1136 }
1137 \define@key{rm3DOptsTopLevel}{3DResources}{%
1138 \def\rmiidOptsTLRes{#1}%

```

This `\define@key`, in turn, executes the family `rm3DOpts`, an original family defined in this package.

```

1139 \setkeys{rm3DOpts}{#1}%
1140 }

```

3DResources. This key recognizes the keys `none`, `foreground`, `background`, and `material`. The keys are optional, and may be repeated more than once. This paragraph processes the keys of the `rm3DOpts` family.

`\rma@ckFileForEmbed` determines if the current resource is already embedded, and defines `\rma@embed` to indicated this, if not embedded, it defines the `rmFileStrm#1`, and defines `\rm@irfstrm` for later use.

```

1141 \def\rma@ckFileForEmbed#1#2{%
1142   \edef\rma@fs@expand{rmFileStrm#1}%
1143   \@ifundefined{\rma@fs@expand}{%
1144     \rm@csarg\xdef{\rma@fs@expand}{%
1145       rmfstream\therm@Cnt-#2#1}\def\rma@embed{1}}%
1146     {\def\rma@embed{0}}%
1147   \edef\rm@irfstrm{\@nameuse{rmFileStrm#1}}%
1148 }

```

none (3D) The none key

```

1149 \define@key{rm3DOpts}{none}{%

```

We use nested key-values, the none key calls the family `rm3DOpts@no`, defined further below.

```

1150   \setkeys{rm3DOpts@no}{rName,#1}%

```

Parse the symbolic name, by using `\filename@parse`.

```

1151   \rma@edefexecute{\noexpand
1152     \filename@parse{\rma@useNamedPath{\rmiiiDOpts@no@rName}}}%
1153   \rma@ckFileForEmbed{\rmiiiDOpts@no@rName}{NONE}%

```

Add to the instance array, file specs/stream, and name tree.

```

1154   \edef\rma@Instances{\rma@Instances
1155     \ps@mark{rmInstances\therm@Cnt} %
1156     {rmInstance\therm@Cnt_NONE\rmiiiDOpts@no@rName}%
1157     /APPEND pdfmark^^J%
1158     \ps@mark/_objdef{rmInstance\therm@Cnt_NONE%
1159       \rmiiiDOpts@no@rName}/type/dict/OBJ pdfmark^^J%
1160     \ps@mark{rmInstance\therm@Cnt_NONE\rmiiiDOpts@no@rName}<<%
1161       /Asset {rmfilespec\therm@Cnt_NONE\rmiiiDOpts@no@rName}%
1162       /Type/RichMediaInstance>>/PUT pdfmark^^J%
1163     \rm@appendFileSpecs{NONE\rmiiiDOpts@no@rName}%
1164     {\filename@area}{\filename@base.\filename@ext}%
1165     {\rma@embed}{\rm@irfstrm}{}%
1166   }%
1167   \edef\rma@appendToNameTree{\rma@appendToNameTree
1168     \rm@appendNameTree{NONE\rmiiiDOpts@no@rName}%
1169     {\filename@area}{\filename@base.\filename@ext}%
1170     {\rma@embed}{\rm@irfstrm}{}%
1171   }%
1172 }

```

foreground (3D) The foreground key

```

1173 \define@key{rm3DOpts}{foreground}{%
1174   \setkeys{rm3DOpts@fg}{rName,flashvars,#1}%

```

Parse the symbolic name, by using `\filename@parse`.

```

1175   \rma@edefexecute{\noexpand
1176     \filename@parse{\rma@useNamedPath{\rmiiiDOpts@fg@rName}}}%
1177   \rma@ckFileForEmbed{\rmiiiDOpts@fg@rName}{FG}%

```

Add to the instance array, file specs/stream, and name tree.

```

1178 \edef\rma@Instances{\rma@Instances
1179 \ps@mark{rmInstances\therm@Cnt} %
1180 {rmInstance\therm@Cnt_FG\rmiiiDOpts@fg@rName}%
1181 /APPEND pdfmark^^J%
1182 \ps@mark/_objdef{rmInstance\therm@Cnt_FG%
1183 \rmiiiDOpts@fg@rName}/type/dict/OBJ pdfmark^^J%
1184 \ps@mark{rmInstance\therm@Cnt_FG\rmiiiDOpts@fg@rName}<<%
1185 /Asset {rmfilespec\therm@Cnt-FG\rmiiiDOpts@fg@rName}%
1186 /Params <</Binding/Foreground%
1187 /FlashVars(\rmiiiDOpts@fg@flashvars)>>%
1188 /Type/RichMediaInstance%
1189 >>/PUT pdfmark^^J%
1190 \rm@appendFileSpecs{FG\rmiiiDOpts@fg@rName}{\filename@area}%
1191 {\filename@base.\filename@ext}{\rma@embed}%
1192 {\rm@irfstrm}{}}%
1193 }%
1194 \edef\rma@appendToNameTree{\rma@appendToNameTree
1195 \rm@appendNameTree{FG\rmiiiDOpts@fg@rName}{\filename@area}%
1196 {\filename@base.\filename@ext}{\rma@embed}%
1197 {\rm@irfstrm}{}}%
1198 }%
1199 }

```

background (3D) **The background key**

```

1200 \define@key{rm3DOpts}{background}{%
1201 \setkeys{rm3DOpts@bg}{rName,flashvars,#1}%
1202 \rma@edefexecute{\noexpand
1203 \filename@parse{\rma@useNamedPath{\rmiiiDOpts@bg@rName}}}%
1204 \rma@ckFileForEmbed{\rmiiiDOpts@bg@rName}{BG}%

```

Add to the instance array, file specs/stream, and name tree.

```

1205 \edef\rma@Instances{\rma@Instances
1206 \ps@mark{rmInstances\therm@Cnt} %
1207 {rmInstance\therm@Cnt_BG\rmiiiDOpts@bg@rName}%
1208 /APPEND pdfmark^^J%
1209 \ps@mark/_objdef{rmInstance\therm@Cnt_BG%
1210 \rmiiiDOpts@bg@rName}/type/dict/OBJ pdfmark^^J%
1211 \ps@mark{rmInstance\therm@Cnt_BG\rmiiiDOpts@bg@rName}<<%
1212 /Asset {rmfilespec\therm@Cnt-BG\rmiiiDOpts@bg@rName}%
1213 /Params <</Binding/Background%
1214 /FlashVars(\rmiiiDOpts@bg@flashvars)>>%
1215 /Type/RichMediaInstance%
1216 >>/PUT pdfmark^^J%
1217 \rm@appendFileSpecs{BG\rmiiiDOpts@bg@rName}{\filename@area}%
1218 {\filename@base.\filename@ext}{\rma@embed}%
1219 {\rm@irfstrm}{}}%
1220 }%

```

```

1221 \edef\rma@appendToNameTree{\rma@appendToNameTree
1222   \rm@appendNameTree{BG\rmiiiDOpts@bg@rName}{\filename@area}%
1223   {\filename@base.\filename@ext}{\rma@embed}%
1224   {\rm@irfstrm}{}}%
1225 }%
1226 }

```

material (3D) This key binds a resource to a material. The resource name is `rName` (as defined by `\saveNamedPath`), the key `mName` is the name of the material the resource is to be bound to; `flashvars` is used to pass variables to the SWF application.

```

1227 \define@key{rm3DOpts}{material}{%
1228   \setkeys{rm3DOpts@mat}{rName,mName,flashvars,#1}%
   Parse the symbolic name, by using \filename@parse.
1229   \rma@edefexecute{\noexpand
1230     \filename@parse{\rma@useNamedPath{\rmiiiDOpts@mat@rName}}}%
1231   \rma@ckFileForEmbed{\rmiiiDOpts@mat@rName}{MAT}%

```

Add to the instance array, file specs/stream, and name tree.

```

1232 \edef\rma@Instances{\rma@Instances
1233   \ps@mark{rmInstances\therm@Cnt} %
1234   {rmInstance\therm@Cnt_MAT\rmiiiDOpts@mat@rName}%
1235   /APPEND pdfmark^^J%
1236   \ps@mark/_objdef{rmInstance\therm@Cnt_MAT%
1237     \rmiiiDOpts@mat@rName}/type/dict/OBJ pdfmark^^J%
1238   \ps@mark{rmInstance\therm@Cnt_MAT\rmiiiDOpts@mat@rName}<<%
1239   /Asset {rmfilespec\therm@Cnt-MAT\rmiiiDOpts@mat@rName}%
1240   /Params <</Binding/Material%
1241     /BindingMaterialName(\rmiiiDOpts@mat@mName)%
1242     /FlashVars(\rmiiiDOpts@mat@flashvars)>>%
1243   /Type/RichMediaInstance%
1244   >>/PUT pdfmark^^J%
1245   \rm@appendFileSpecs{MAT\rmiiiDOpts@mat@rName}{\filename@area}%
1246   {\filename@base.\filename@ext}{\rma@embed}%
1247   {\rm@irfstrm}{}}%
1248 }%
1249 \edef\rma@appendToNameTree{\rma@appendToNameTree
1250   \rm@appendNameTree{MAT\rmiiiDOpts@mat@rName}{\filename@area}%
1251   {\filename@base.\filename@ext}{\rma@embed}%
1252   {\rm@irfstrm}{}}%
1253 }%
1254 }

```

Process none values

```

1255 \define@key{rm3DOpts@no}{rName}[]{\def\rmiiiDOpts@no@rName{#1}}

```

Process foreground values

```

1256 \define@key{rm3DOpts@fg}{rName}[]{\def\rmiiiDOpts@fg@rName{#1}}
1257 \define@key{rm3DOpts@fg}{flashvars}[]{\def\rmiiiDOpts@fg@flashvars{#1}}

```

Process background values

```
1258 \define@key{rm3D0pts@bg}{rName}[]{\def\rmiid0pts@bg@rName{#1}}
1259 \define@key{rm3D0pts@bg}{flashvars}[]{\def\rmiid0pts@bg@flashvars{#1}}
```

Process material values

```
1260 \define@key{rm3D0pts@mat}{rName}[]{\def\rmiid0pts@mat@rName{#1}}
1261 \define@key{rm3D0pts@mat}{mName}[]{\def\rmiid0pts@mat@mName{#1}}
1262 \define@key{rm3D0pts@mat}{flashvars}[]{%
1263   \def\rmiid0pts@mat@flashvars{#1}}
```

4.4.1 Code from movie15

My gracious thanks to Alexander Grahn for granting permission to use some of his movie15 code.

```
1264 \newread\@MXV@@viewsfile% file handle for views file
1265 \newboolean{@MXV@eof}%
1266 \newcount\@MXV@viewscount%counter for number of 3D views per inclusion
1267 \newboolean{@MXV@viewsprovided}%3d views file provided?
1268 \newboolean{@MXV@defaultviewprovided}%default 3D view provided?
1269 \newcount\@MXV@nodecount% number of node dicts
1270 \newcount\@MXV@cscount% number of cross section dicts
```

Default values

```
1271 \def\@MXV@aac{30}% aperture angle of camera
1272 \def\@MXV@roll{0}% camera roll angle
1273 \def\@MXV@defaultbg{1 1 1}%
1274 \def\@MXV@background{/BG<</CS/DeviceRGB/C[1 1 1]>>}%
1275 \def\@MXV@defaultlights{}%
1276 \def\@MXV@lights{}%
1277 \def\@MXV@defaultrender{Solid}%
1278 \def\@MXV@render{/RM <</Subtype/Solid>>}%
1279 \def\@MXV@naentry{}% %takes array of Node dicts
1280 \def\@MXV@saentry{}% %takes array of cross section dicts
1281 \let\@MXV@jscriptiid\@empty
1282 \let\@rma@rmAnnot@iiiDjs\@empty
1283 \let\@MXV@varray\@empty
1284 \let\@additional@Instances\@empty
1285 \def\@MXV@defaultview{}
1286 \def\@MXV@coo{0 0 0}% centre of orbit
1287 \def\@MXV@ctoc{0 -1 0}% centre of orbit to camera vector
1288 \def\@MXV@roo{0}% radius of orbit
1289 \def\@MXV@viewsfileii{}%file of views of the 3D object (new format)
1290 \setboolean{@MXV@viewsprovided}{false}%
1291 \setboolean{@MXV@defaultviewprovided}{false}%
1292 \def\@MXV@iidview{}%
```

\@MXV@ciwmatrix is a macro for building the transformation matrix

#1,#2,#3 centre of orbit coordinates (coo)

#4,#5,#6 centre of orbit to camera direction vector (c2c)

```

#7 orbital radius (roo)

#8 camera roll (roll)

1293 \def\@MXV@ciimatrix#1 #2 #3 #4 #5 #6 #7 #8 {%
    View vector (opposite to c2c)
1294 \FPupn\@MXV@viewx{#4 neg}%
1295 \FPupn\@MXV@viewy{#5 neg}%
1296 \FPupn\@MXV@viewz{#6 neg}%
    Normalize view vector
1297 \FPupn\@MXV@modulo{\@MXV@viewx{}} copy mul %
1298 \@MXV@viewy{}} copy mul + %
1299 \@MXV@viewz{}} copy mul + 2 swap root%
1300 }%
1301 \FPupn\@MXV@viewx{\@MXV@viewx{}} \@MXV@modulo{}} div}%
1302 \FPupn\@MXV@viewy{\@MXV@viewy{}} \@MXV@modulo{}} div}%
1303 \FPupn\@MXV@viewz{\@MXV@viewz{}} \@MXV@modulo{}} div}%
    Camera roll
1304 \FPupn\@MXV@sinroll{#8 180.0 div \FPpi{}} mul sin}%
1305 \FPupn\@MXV@cosroll{#8 180.0 div \FPpi{}} mul cos}%
    Top and bottom views
1306 \FPupn\@MXV@leftx{-1.0}%
1307 \FPupn\@MXV@lefty{0.0}%
1308 \FPupn\@MXV@leftz{0.0}%
1309 \FPifneg\@MXV@viewz% top view
1310 %up-vector
1311 \FPupn\@MXV@upx{0.0}%
1312 \FPupn\@MXV@upy{1.0}%
1313 \FPupn\@MXV@upz{0.0}%
1314 \else% bottom view
1315 %up-vector
1316 \FPupn\@MXV@upx{0.0}%
1317 \FPupn\@MXV@upy{-1.0}%
1318 \FPupn\@MXV@upz{0.0}%
1319 \fi%
1320 \FPupn\@MXV@sumxy{\@MXV@viewx{}} abs \@MXV@viewy{}} abs add}%
1321 \FPifeq\@MXV@sumxy{0}\else% other views than top and bottom
1322 %up-vector = up_world - (up_world dot view) view
1323 \FPupn\@MXV@upx{\@MXV@viewz{}} \@MXV@viewx{}} mul neg}%
1324 \FPupn\@MXV@upy{\@MXV@viewz{}} \@MXV@viewy{}} mul neg}%
1325 \FPupn\@MXV@upz{\@MXV@viewz{}} \@MXV@viewz{}} mul neg 1.0 add}%
1326 %normalize up-vector
1327 \FPupn\@MXV@modulo{\@MXV@upx{}} copy mul \@MXV@upy{}} copy %
1328 mul + \@MXV@upz{}} copy mul + 2 swap root}%
1329 \FPupn\@MXV@upx{\@MXV@upx{}} \@MXV@modulo{}} div}%
1330 \FPupn\@MXV@upy{\@MXV@upy{}} \@MXV@modulo{}} div}%
1331 \FPupn\@MXV@upz{\@MXV@upz{}} \@MXV@modulo{}} div}%
1332 %left vector = up x view

```

```

1333 \FPupn\@MXV@leftx{\@MXV@viewz{} \@MXV@upy{} mul %
1334 \@MXV@viewy{} \@MXV@upz{} mul sub}%
1335 \FPupn\@MXV@lefty{\@MXV@viewx{} \@MXV@upz{} mul %
1336 \@MXV@viewz{} \@MXV@upx{} mul sub}%
1337 \FPupn\@MXV@leftz{\@MXV@viewy{} \@MXV@upx{} mul %
1338 \@MXV@viewx{} \@MXV@upy{} mul sub}%
1339 %normalize left vector
1340 \FPupn\@MXV@modulo{\@MXV@leftx{} copy mul \@MXV@lefty{} %
1341 copy mul + \@MXV@leftz{} copy mul + 2 swap root}%
1342 \FPupn\@MXV@leftx{\@MXV@leftx{} \@MXV@modulo{} div}%
1343 \FPupn\@MXV@lefty{\@MXV@lefty{} \@MXV@modulo{} div}%
1344 \FPupn\@MXV@leftz{\@MXV@leftz{} \@MXV@modulo{} div}%
1345 \fi%

```

Apply camera roll

```

1346 \FPupn\@MXV@leftxprime{\@MXV@leftx{} \@MXV@cosroll{} mul %
1347 \@MXV@upx{} \@MXV@sinroll{} mul +}%
1348 \FPupn\@MXV@leftyprime{\@MXV@lefty{} \@MXV@cosroll{} mul %
1349 \@MXV@upy{} \@MXV@sinroll{} mul +}%
1350 \FPupn\@MXV@leftzprime{\@MXV@leftz{} \@MXV@cosroll{} mul %
1351 \@MXV@upz{} \@MXV@sinroll{} mul +}%
1352 \FPupn\@MXV@upxprime{\@MXV@upx{} \@MXV@cosroll{} mul %
1353 \@MXV@leftx{} \@MXV@sinroll{} mul sub}%
1354 \FPupn\@MXV@upyprime{\@MXV@upy{} \@MXV@cosroll{} mul %
1355 \@MXV@lefty{} \@MXV@sinroll{} mul sub}%
1356 \FPupn\@MXV@upzprime{\@MXV@upz{} \@MXV@cosroll{} mul %
1357 \@MXV@leftz{} \@MXV@sinroll{} mul sub}%
1358 \FPupn\@MXV@leftx{\@MXV@leftxprime}%
1359 \FPupn\@MXV@lefty{\@MXV@leftyprime}%
1360 \FPupn\@MXV@leftz{\@MXV@leftzprime}%
1361 \FPupn\@MXV@upx{\@MXV@upxprime}%
1362 \FPupn\@MXV@upy{\@MXV@upyprime}%
1363 \FPupn\@MXV@upz{\@MXV@upzprime}%

```

Translation vector

```

1364 \FPupn\@MXV@roo{#7 abs}%
1365 \FPifeq\@MXV@roo{0}\FPupn\@MXV@roo{0.0000001}\fi%
1366 \FPupn\@MXV@transx{#1 \@MXV@roo{} \@MXV@viewx{} mul sub}%
1367 \FPupn\@MXV@transy{#2 \@MXV@roo{} \@MXV@viewy{} mul sub}%
1368 \FPupn\@MXV@transz{#3 \@MXV@roo{} \@MXV@viewz{} mul sub}%

```

Rotation matrix

```

1369 \xdef\@MXV@matrix{%
1370 \@MXV@leftx\space\@MXV@lefty\space\@MXV@leftz\space%
1371 \@MXV@upx\space\@MXV@upy\space\@MXV@upz\space%
1372 \@MXV@viewx\space\@MXV@viewy\space\@MXV@viewz}%

```

Transformation matrix

```

1373 \xdef\@MXV@matrix{%
1374 \@MXV@matrix\space\@MXV@transx\space%
1375 \@MXV@transy\space\@MXV@transz%

```

```

1376 }%
1377 }% end of \@MXV@ciwmatrix

Macro for parsing one line of 3D views file (old format)
1378 \newcommand{\@MXV@parseline}[6] []{%
1379 \pdfstringdef\@MXV@xname{#1}% name of the view (optional)
1380 \ifthenelse{\equal{#2}{}}{%
1381 \xdef\@MXV@coo{0 0 0}%
1382 }{%
1383 \xdef\@MXV@coo{#2}%
1384 }%
1385 \ifthenelse{\equal{#3}{}}{%
1386 \xdef\@MXV@ctoc{0 -1 0}%
1387 }{%
1388 \xdef\@MXV@ctoc{#3}%
1389 }%
1390 \ifthenelse{\equal{#4}{}}{%
1391 \xdef\@MXV@roo{0}%
1392 }{%
1393 \xdef\@MXV@roo{#4}%
1394 }%
1395 \ifthenelse{\equal{#5}{}}{%
1396 \xdef\@MXV@roll{0}%
1397 }{%
1398 \xdef\@MXV@roll{#5}%
1399 }%
1400 \ifthenelse{\equal{#6}{}}{%
1401 \xdef\@MXV@aac{30}%
1402 }{%
1403 \xdef\@MXV@aac{#6}%
1404 }%
1405 }

For parsing lines of views file (new format)
1406 \define@key{MXV@view}{VIEW} []{%
1407 \ifnum\@MXV@cursection<\z@else%
1408 \PackageError{rmanot}{%
1409 File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1410 A VIEW section cannot be nested into another section}{}%
1411 \fi%
1412 \def\@MXV@cursection{0}%
1413 \pdfstringdef\@MXV@xname{#1}% name of the view (optional)
1414 %default camera settings
1415 \gdef\@MXV@coo{0 0 0}%
1416 \gdef\@MXV@ctoc{0 -1 0}%
1417 \gdef\@MXV@roo{0}%
1418 \gdef\@MXV@roll{0}%
1419 \gdef\@MXV@aac{30}%
1420 %default background, lights, render mode
1421 \xdef\@MXV@background{BG <</CS/DeviceRGB/C [\@MXV@defaultbg]>>}%
1422 \xdef\@MXV@lights{LS <</Subtype/\@MXV@defaultlights>>}%

```

```

1423 \gdef\@MXV@render{/RM <</Subtype/\@MXV@defaultrender>>}}%
1424 %initialise array of node dicts
1425 \gdef\@MXV@naarray{}%
1426 \global\@MXV@nodecount=\z@
1427 %initialise array of crosssection dicts
1428 \gdef\@MXV@saarray{}%
1429 \global\@MXV@cscount=\z@
1430 }
1431 \define@key{MXV@view}{COO}{%
1432 \ifnum\@MXV@cursection=\z@\else%
1433 \PackageError{rmannot}{%
1434 File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1435 COO entry not allowed here; must go into a VIEW section}{}%
1436 \fi%
1437 \xdef\@MXV@coo{#1}%
1438 }
1439 \define@key{MXV@view}{C2C}{%
1440 \ifnum\@MXV@cursection=\z@\else%
1441 \PackageError{rmannot}{%
1442 File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1443 C2C entry not allowed here; must go into a VIEW section}{}%
1444 \fi\xdef\@MXV@ctoc{#1}%
1445 }
1446 \define@key{MXV@view}{ROO}{%
1447 \ifnum\@MXV@cursection=\z@\else%
1448 \PackageError{rmannot}{%
1449 File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1450 ROO entry not allowed here; must go into a VIEW section}{}%
1451 \fi\xdef\@MXV@roo{#1}%
1452 }
1453 \define@key{MXV@view}{AAC}{%
1454 \ifnum\@MXV@cursection=\z@\else%
1455 \PackageError{rmannot}{%
1456 File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1457 AAC entry not allowed here; must go into a VIEW section}{}%
1458 \fi\xdef\@MXV@aac{#1}%
1459 }
1460 \define@key{MXV@view}{ROLL}{%
1461 \ifnum\@MXV@cursection=\z@\else%
1462 \PackageError{rmannot}{%
1463 File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1464 ROLL entry not allowed here; must go into a VIEW section}{}%
1465 \fi\xdef\@MXV@roll{#1}%
1466 }
1467 \define@key{MXV@view}{BGCOLOR}{%
1468 \ifnum\@MXV@cursection=\z@\else%
1469 \PackageError{rmannot}{%
1470 File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1471 BGOLOR entry not allowed here; must go into a VIEW section}{}%
1472 \fi\def\@MXV@background{BG<</CS/DeviceRGB/C[#1]>>}}%

```

```

1473 }
1474 \define@key{MXV@view}{LIGHTS}{%
1475   \ifnum\MXV@cursection=\z@ \else%
1476     \PackageError{rmannot}{%
1477       File \MXV@viewsfileii, line \the\MXV@inputlineno: %
1478       LIGHTS entry not allowed here; must go into a VIEW section}{}%
1479   \fi\def\MXV@lights{/LS <</Subtype/#1>>}%
1480 }
1481 \define@key{MXV@view}{RENDERMODE}{%
1482   \ifnum\MXV@cursection=\z@
1483     \def\MXV@render{/RM <</Subtype/#1>>}%
1484   \else%
1485     \ifnum\MXV@cursection=\@ne
1486       \def\MXV@nrender{/RM <</Subtype/#1>>}%
1487     \else%
1488       \PackageError{rmannot}{%
1489         File \MXV@viewsfileii, line \the\MXV@inputlineno: %
1490         RENDERMODE entry not allowed here; must go into %
1491         either a VIEW or a PART section}{}%
1492     \fi%
1493   \fi%
1494 }
1495 \define@key{MXV@view}{PART}[]{%
1496   \ifnum\MXV@cursection=\z@ \else%
1497     \PackageError{rmannot}{%
1498       File \MXV@viewsfileii, line \the\MXV@inputlineno: %
1499       PART not allowed here; must be a sub-section of %
1500       a VIEW section}{}%
1501   \fi%
1502   \ifthenelse{\equal{#1}{}}{%
1503     \PackageError{rmannot}{%
1504       File \MXV@viewsfileii, line \the\MXV@inputlineno: %
1505       You must provide a valid PART name (PART=<part name>), as
1506       shown in the model tree of the 3D object %
1507       (go to 'View' -> 'Navigation Panels' -> 'Model Tree' %
1508       in Adobe Reader)}{}%
1509   }{%
1510     \def\MXV@cursection{1}%
1511     \pdfstringdef\MXV@partname{#1}% name of the part
1512     \gdef\MXV@nopacity{}%
1513     \gdef\MXV@nvisibility{}%
1514     \gdef\MXV@nrender{}%
1515     \gdef\MXV@ntransform{}%
1516   }
1517 \define@key{MXV@view}{CROSSSECT}[]{%
1518   \ifnum\MXV@cursection=\z@ \else%
1519     \PackageError{rmannot}{%
1520       File \MXV@viewsfileii, line \the\MXV@inputlineno: %
1521       CROSSSECT not allowed here; must be a sub-section of %
1522       a VIEW section}{}%

```

```

1523 \fi%
1524 \ifthenelse{\equal{#1}{}}{}{}{%
1525   \PackageWarning{rmannot}{%
1526     File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1527     CROSSECT does not take a value%
1528   }%
1529 }%
1530 \def\@MXV@cursection{2}%
1531 \gdef\@MXV@cscenter{0 0 0}%
1532 \gdef\@MXV@csorient{null 0 0}%
1533 }
1534 \define@key{MXV@view}{OPACITY}{%
1535   \ifnum\@MXV@cursection=\@one\else%
1536     \PackageError{rmannot}{%
1537       File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1538       OPACITY entry not allowed here; must go into a PART section}{}%
1539   \fi%
1540   \gdef\@MXV@nopacity{/O #1}%
1541 }
1542 \define@key{MXV@view}{VISIBLE}{%
1543   \ifnum\@MXV@cursection=\@one\else%
1544     \PackageError{rmannot}{%
1545       File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1546       VISIBLE entry not allowed here; must go into a PART section}{}%
1547   \fi%
1548   \gdef\@MXV@nvisibility{/V #1}%
1549 }
1550 \define@key{MXV@view}{MATRIX}{%
1551   \ifnum\@MXV@cursection=\@one\else%
1552     \PackageError{rmannot}{%
1553       File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1554       MATRIX entry not allowed here; must go into a PART section}{}%
1555   \fi%
1556   \gdef\@MXV@ntransform{/M [#1]}%
1557 }
1558 \define@key{MXV@view}{CENTER}{%
1559   \ifnum\@MXV@cursection=2\relax\else%
1560     \PackageError{rmannot}{%
1561       File \@MXV@viewsfileii, line \the\@MXV@inputlineno: CENTER %
1562       entry not allowed here; must go into a CROSSECT section}{}%
1563   \fi%
1564   \gdef\@MXV@cscenter{#1}%
1565 }
1566 \define@key{MXV@view}{ORIENTATION}{%
1567   \ifnum\@MXV@cursection=2\relax\else%
1568     \PackageError{rmannot}{%
1569       File \@MXV@viewsfileii, line \the\@MXV@inputlineno: ORIENT %
1570       entry not allowed here; must go into a CROSSECT section}{}%
1571   \fi%
1572   \gdef\@MXV@csorient{#1}%

```

```

1573 }
1574 \define@key{MXV@view}{END}[]{}%
1575 \ifcase\@MXV@cursection%
1576   %END VIEW
1577   \edef\@MXV@args{%
1578     \@MXV@coo\space\@MXV@ctoc\space\@MXV@roo\space%
1579     \@MXV@roll\space}%
1580   \expandafter\@MXV@ciwmatrix\@MXV@args% build C2W matrix
1581   \global\advance\@MXV@viewscount by \@ne%
1582   \ifthenelse{\equal{\@MXV@xname}{}}{}{}% default view name
1583   \pdfstringdef\@MXV@xname{View \the\@MXV@viewscount}%
1584   }{}%
1585   \ifthenelse{\equal{\@MXV@naarray}{}}{}{}{}%
1586   \gdef\@MXV@naentry{/NR true/NA [\@MXV@naarray]}%
1587   }%
1588   \ifthenelse{\equal{\@MXV@saarray}{}}{}{}{}%
1589   \gdef\@MXV@saentry{/SA [\@MXV@saarray]}%
1590   }%
1591   \@MXV@viewobj% create pdf object of 3D view
1592   %append current view obj ref to VA array
1593   \xdef\@MXV@varray{\@MXV@varray\space\@MXV@@viewobj}%
1594   \global\@MXV@viewsprovidedtrue%
1595   \def\@MXV@cursection{-1}%
1596 \or%
1597   %END PART
1598   \global\advance\@MXV@nodecount by \@ne
1599   \@MXV@nodeobj% create pdf object of 3D node dict
1600   %append it to node array
1601   \xdef\@MXV@naarray{\@MXV@naarray\space\@MXV@@nodeobj}%
1602   \def\@MXV@cursection{0}%
1603 \or%
1604   %END CROSSSECT
1605   \global\advance\@MXV@cscount by \@ne
1606   \@MXV@csoobj% create pdf object of 3D cross section dict
1607   %append it to cross section array
1608   \xdef\@MXV@saarray{\@MXV@saarray\space\@MXV@@csoobj}%
1609   \def\@MXV@cursection{0}%
1610 \else%
1611   \PackageError{rmannot}{}%
1612     File \@MXV@viewsfileii, line \the\@MXV@inputlineno: %
1613     There is nothing to be ENDED here}{}%
1614 \fi%
1615 }%

Macro for generating an array of 3D views (varray)
1616 \def\@MXV@procinputline#1{\setkeys{MXV@view}{#1}}
1617 \newcount\@MXV@inputlineno
1618 \def\@MXV@buildva{%
1619   \global\@MXV@viewscount=0\relax% dps
1620   \xdef\@MXV@varray{}% empty varray

```

```

1621 %
1622 %default view (one of the command options 3Dcoo, 3Dc2c, etc. given)
1623 \edef\@MXV@args{%
1624   \@MXV@coo\space\@MXV@ctoc\space\@MXV@roo\space%
1625   \@MXV@roll\space}%
1626 \expandafter\@MXV@ciwmatrix\@MXV@args% build C2W matrix
1627 \pdfstringdef\@MXV@xname{Default}%
1628 \if@MXV@defaultviewprovided%
1629   \@MXV@viewobj% create pdf object of 3D view
1630 % \edef\@MXV@defaultview{/3DV \@MXV@@viewobj}%
1631 \edef\@MXV@defaultview{\@MXV@@viewobj}%
1632 \fi%

```

Read out 3D views file (new version)

```

1633 \def\@MXV@cursection{-1}% views file is divided in sections
1634 \IfFileExists{\@MXV@viewsfileii}{%
1635   \begingroup%
1636   \endlinechar=-1% suppress trailing space at input line end
1637   \@MXV@inputlineno=\z@%
1638   \openin\@MXV@@viewsfile=\@MXV@viewsfileii%
1639   \read\@MXV@@viewsfile to \@MXV@inputline%
1640   \ifeof\@MXV@@viewsfile\setboolean{\@MXV@eof}{true}\else%
1641     \setboolean{\@MXV@eof}{false}\fi%
1642   \whiledo{\not\boolean{\@MXV@eof}}{%
1643     \advance\@MXV@inputlineno by \@one%
1644     %process input line
1645     \edef\@MXV@@inputline{\@MXV@inputline}%
1646     \expandafter\@MXV@procinputline\@MXV@@inputline%
1647     \read\@MXV@@viewsfile to \@MXV@inputline%
1648     \ifeof\@MXV@@viewsfile%
1649       \setboolean{\@MXV@eof}{true}%
1650     \else%
1651       \setboolean{\@MXV@eof}{false}%
1652     \fi%
1653   }%
1654   \closein\@MXV@@viewsfile%
1655   \endgroup%
1656 }{}%

```

Make the first view in the VA array the default view, if no default one has explicitly been provided, but if the VA array itself is empty too (no additional views provided) use our fallback view (c2c=0 -1 0) as default

```

1657 \ifthenelse{\not\boolean{\@MXV@defaultviewprovided}}%
1658   \and\boolean{\@MXV@viewsprovided}}{%
1659 % \xdef\@MXV@defaultview{/3DV/F}%
1660 \xdef\@MXV@defaultview{/F}%
1661 }{}%
1662 \ifthenelse{\not\boolean{\@MXV@defaultviewprovided}}%
1663   \and\not\boolean{\@MXV@viewsprovided}}{%
1664   \@MXV@viewobj% create pdf object of 3D view

```

```

1665 %      \edef\@MXV@defaultview{/3DV \@MXV@@viewobj}%
1666      \edef\@MXV@defaultview{\@MXV@@viewobj}%
1667    }{}%
1668 }

```

Following macros, including the 3D inclusion macro have driver specific implementations dvips versions macro for creating 3D view object and associated projection dict

```

1669 \def\@MXV@viewobj{\literalps@out{%
1670 %projection dict
1671 \ps@mark/_objdef {pdict\therm@Cnt_\the\@MXV@viewscount}%
1672 /type/dict/OBJ pdfmark^^J%
1673 \ps@mark{pdict\therm@Cnt_\the\@MXV@viewscount} <<%
1674 /Subtype/P/FOV \@MXV@aac/PS/Min>>/PUT pdfmark^^J%
1675 \ps@mark/_objdef {viewobj\therm@Cnt_\the\@MXV@viewscount}%
1676 /type/dict/OBJ pdfmark^^J%
1677 \ps@mark{viewobj\therm@Cnt_\the\@MXV@viewscount} <<%
1678 /MS/M%
1679 /CO \@MXV@roo%
1680 /P {pdict\therm@Cnt_\the\@MXV@viewscount}%
1681 /C2W[\@MXV@matrix]%
1682 /XN(\@MXV@xname)%
1683 /IN(\@MXV@xname)%
1684 \@MXV@background%
1685 \@MXV@lights%
1686 \@MXV@render%
1687 \@MXV@naentry%
1688 \@MXV@saentry%
1689 >>%
1690 /PUT pdfmark
1691 }%
1692 \xdef\@MXV@@viewobj{{viewobj\therm@Cnt_\the\@MXV@viewscount}}%
1693 }%
1694 %3D node object
1695 \def\@MXV@nodeobj{\literalps@out{%
1696 \ps@mark/type/dict%
1697 /_objdef {nodeobj\therm@Cnt_\the\@MXV@viewscount_%
1698 \the\@MXV@nodecount}/OBJ pdfmark^^J%
1699 \ps@mark{nodeobj\therm@Cnt_\the\@MXV@viewscount_%
1700 \the\@MXV@nodecount}<<%
1701 /Type/3DNode%
1702 /N (\@MXV@partname)%
1703 \@MXV@nopacity\@MXV@nvisibility\@MXV@ntransform%
1704 \@MXV@nrender%
1705 >>/PUT pdfmark
1706 }%
1707 \xdef\@MXV@@nodeobj{%
1708 {nodeobj\therm@Cnt_\the\@MXV@viewscount_%
1709 \the\@MXV@nodecount}}%
1710 }%

```

```

1711 %3D cross section object
1712 \def\@MXV@csobj{\literalps@out{%
1713   \ps@mark/type/dict%
1714     /_objdef {csobj\therm@Cnt_\the\@MXV@viewscount_%
1715     \the\@MXV@cscount}/OBJ pdfmark^^J%
1716   \ps@mark{csobj\therm@Cnt_\the\@MXV@viewscount_%
1717     \the\@MXV@cscount}<<%
1718     /Type/3DCrossSection%
1719     /C [\@MXV@cscenter]%
1720     /O [\@MXV@csorient]%
1721   >>/PUT pdfmark
1722 }%
1723 \xdef\@MXV@@csobj{%
1724   {csobj\therm@Cnt_\the\@MXV@viewscount_\the\@MXV@cscount}}%
1725 }%

```

MXV@user family

```

1726 \define@key{MXV@user}{3Dbg}[1 1 1]{%
1727   \def\@MXV@defaultbg{#1}%
1728   \def\@MXV@background{/BG<</CS/DeviceRGB/C[#1]>>}%
1729 }
1730 \define@key{MXV@user}{3Djscript}{%
1731   \def\rma@rmAnnot@iiidjs{#1}%
1732   \ifx\rma@rmAnnot@iiidjs\@empty\let\rma@addResources\@empty
1733   \let\rma@addFileSpecs\@empty\else

```

We process resources when there are some to process :-)

```

1734   \rma@toks={}\def\rmiid@addToScriptsArray{%
1735     \@for\rma@arg:=\rma@rmAnnot@iiidjs\do{%
1736       \rma@edefexecute{\noexpand
1737         \filename@parse{\rma@useNamedPath{\rma@arg}}}%
1738       \@ifundefined{filename@ext}{%
1739         \rma@PkEr@iii{\rma@useNamedPath{\rma@arg}}}{}%
1740       \edef\rmiid@addToScriptsArray{\rmiid@addToScriptsArray
1741         \ps@mark{jscriptiiid\therm@Cnt}%
1742         {rmfilespec\therm@Cnt-JS\rma@arg}%
1743         /APPEND pdfmark^^J%
1744       }%
1745       \edef\rma@fs@expand{rmFileStrm\rma@arg}%
1746       \@ifundefined{\rma@fs@expand}{%
1747         \rm@csarg\xdef{\rma@fs@expand}%
1748         {rmfstream\therm@Cnt-JS\rma@arg}%
1749         \def\rma@embed{1}}{\def\rma@embed{0}}%
1750       \edef\rma@tmp@exp{\the\rma@toks%
1751         \noexpand\\{JS\rma@arg}%
1752         {\filename@area}{\filename@base.\filename@ext}}%
1753         {\rma@embed}{\csname\rma@fs@expand\endcsname}%
1754         {\rm@csarg\noexpand\rma@mt@\rma@arg}}}%
1755       \rma@toks=\expandafter{\rma@tmp@exp}%
1756     }% do

```

```

1757 \let\\rm@appendNameTree
1758 \expandafter\xdef\expandafter\rma@addResources%
1759 \expandafter{\the\rma@toks}%
1760 \let\\rm@appendFileSpecs
1761 \expandafter\xdef\expandafter\rma@addFileSpecs%
1762 \expandafter{\the\rma@toks}%
1763 \fi
1764 }
1765 \define@key{MXV@user}{3Dcoo}{%
1766 \def\@MXV@coo{#1}%
1767 \setboolean{MXV@defaultviewprovided}{true}%
1768 }
1769 \define@key{MXV@user}{3Dc2c}{%
1770 \def\@MXV@ctoc{#1}%
1771 \setboolean{MXV@defaultviewprovided}{true}%
1772 }
1773 \define@key{MXV@user}{3Droo}{%
1774 \def\@MXV@roo{#1}%
1775 \setboolean{MXV@defaultviewprovided}{true}%
1776 }
1777 \define@key{MXV@user}{3Daac}{%
1778 \def\@MXV@aac{#1}%
1779 \setboolean{MXV@defaultviewprovided}{true}%
1780 }
1781 \define@key{MXV@user}{3Droll}{%
1782 \def\@MXV@roll{#1}%
1783 \setboolean{MXV@defaultviewprovided}{true}%
1784 }

```

Since we are starting fresh, we don't use the old format used by movie15, so I am renaming 3Dviews2 to 3Dviews and eliminating the old format and code completely.

```

1785 \define@key{MXV@user}{3Dviews}{%
1786 \IfFileExists{#1}{%
1787 \def\@MXV@viewsfileii{#1}%
1788 }{%
1789 \PackageError{rmanot}{3D views file '#1' cannot be opened%
1790 }{%
1791 Make sure file '#1' exists and is readable!%
1792 }%
1793 }%
1794 }
1795 \define@choicekey+{MXV@user}{3Dlights}{%
1796 {None,White,Day,Night,Hard,Primary,Blue,%
1797 Red,Cube,CAD,Headlamp}[Cube]{%
1798 \gdef\@MXV@defaultlights{#1}%
1799 \gdef\@MXV@lights{/LS <</Subtype/#1>>}%
1800 }{\PackageWarning{rmanot}{Bad choice for 3Dlights, permissible
1801 values are None, White, Day, Night, Hard, Primary, Blue,
1802 Red, Cube, CAD, HeadLamp. Try again}}

```

```

1803 \define@choicekey+{MXV@user}{3Drender}%
1804   {Solid,SolidWireframe,Transparent,TransparentWireframe,%
1805     BoundingBox,TransparentBoundingBox,TransparentBoundingBoxOutline,%
1806     Wireframe,ShadedWireframe,HiddenWireframe,Vertices,ShadedVertices,%
1807     SolidOutline,Illustration,ShadedIllustration}[Solid]{%
1808   \gdef\@MXV@defaultrender{#1}%
1809   \gdef\@MXV@render{/RM <</Subtype/#1>>}}%
1810 }\PackageWarning{rmanot}{Bad choice for 3Dlights, permissible
1811 values are Solid, SolidWireframe, Transparent, TransparentWireframe,
1812 BoundingBox, TransparentBoundingBox, TransparentBoundingBoxOutline,
1813 Wireframe, ShadedWireframe, HiddenWireframe, Vertices,
1814 ShadedVertices, SolidOutline, Illustration, ShadedIllustration.
1815 Try again}}
1816 </3Dcode>
1817 <*package>
1818 \rma@input@iiidCode
1819 </package>

```

5 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	
\@FndSuppExtfalse	593, 692
\@FndSuppExttrue	696
\@MXV@csobj	1608, 1723
\@MXV@inputline	1645, 1646
\@MXV@nodeobj	1601, 1707
\@MXV@viewobj	1593, 1630, 1631, 1665, 1666, 1692
\@MXV@viewsfile	1264, 1638–1640, 1647, 1648, 1654
\@MXV@aac	1271, 1401, 1403, 1419, 1458, 1674, 1778
\@MXV@args	1577, 1580, 1623, 1626
\@MXV@background	1274, 1421, 1472, 1684, 1728
\@MXV@buildva	669, 1618
\@MXV@ciwmatrix	44, 1293, 1377, 1580, 1626
\@MXV@coo	1286, 1381, 1383, 1415, 1437, 1578, 1624, 1766
\@MXV@cosroll	1305, 1346, 1348, 1350, 1352, 1354, 1356
\@MXV@cscenter	1531, 1564, 1719
\@MXV@cscount	1270, 1429, 1605, 1715, 1717, 1724
\@MXV@csobj	1606, 1712
\@MXV@csorient	1532, 1572, 1720
\@MXV@ctoc	1287, 1386, 1388, 1416, 1444, 1578, 1624, 1770
\@MXV@cursection	1407, 1412, 1432, 1440, 1447, 1454, 1461, 1468, 1475, 1482, 1485, 1496, 1510, 1518, 1530, 1535, 1543, 1551, 1559, 1567, 1575, 1595, 1602, 1609, 1633
\@MXV@defaultbg	1273, 1421, 1727
\@MXV@defaultlights	1275, 1422, 1798
\@MXV@defaultrender	1277, 1423, 1808
\@MXV@defaultview	924, 925, 1285, 1630, 1631, 1659, 1660, 1665, 1666
\@MXV@getlabelvalue	199, 209, 212, 214, 227
\@MXV@iidview	1292
\@MXV@inputline	1639, 1645, 1647
\@MXV@inputlineno	1409, 1434, 1442, 1449, 1456, 1463, 1470, 1477, 1489, 1498, 1504, 1520, 1526, 1537, 1545, 1553, 1561, 1569, 1612, 1617, 1637, 1643
\@MXV@jscriptiid	658, 665, 927, 928, 1281
\@MXV@labeltoaux	206, 754
\@MXV@leftx	1306, 1333, 1340, 1342, 1346, 1353, 1358, 1370
\@MXV@leftxprime	1346, 1358
\@MXV@lefty	1307, 1335, 1340, 1343, 1348, 1355, 1359, 1370
\@MXV@leftyprime	1348, 1359
\@MXV@leftz	1308, 1337, 1341, 1344, 1350, 1357, 1360, 1370
\@MXV@leftzprime	1350, 1360
\@MXV@lights	1276, 1422, 1479, 1685, 1799
\@MXV@matrix	1369, 1373, 1374, 1681
\@MXV@modulo	1297, 1301–1303, 1327, 1329–1331, 1340, 1342–1344
\@MXV@naarray	1425, 1585, 1586, 1601
\@MXV@naentry	1279, 1586, 1687
\@MXV@newlabel	197, 208, 209, 753
\@MXV@nodecount	1269, 1426, 1598, 1698, 1700, 1709
\@MXV@nodeobj	1599, 1695
\@MXV@nopacity	1512, 1540, 1703
\@MXV@nrender	1486, 1514, 1704
\@MXV@ntransform	1515, 1556, 1703
\@MXV@nvisibility	1513, 1548, 1703
\@MXV@parseline	1378
\@MXV@partname	1511, 1702
\@MXV@procinputline	1616, 1646
\@MXV@render	1278, 1423, 1483, 1686, 1809
\@MXV@roll	1272, 1396, 1398, 1418, 1465, 1579, 1625, 1782
\@MXV@roo	1288, 1364–1368, 1391, 1393, 1417, 1451, 1578, 1624, 1679, 1774
\@MXV@saarray	1428, 1588, 1589, 1608
\@MXV@saentry	1280, 1589, 1688
\@MXV@sinroll	1304, 1347, 1349, 1351, 1353, 1355, 1357
\@MXV@sumxy	1320, 1321
\@MXV@transx	1366, 1374
\@MXV@transy	1367, 1375
\@MXV@transz	1368, 1375
\@MXV@upx	1311, 1316, 1323, 1327, 1329, 1336, 1337, 1347, 1352, 1361, 1371
\@MXV@upxprime	1352, 1361
\@MXV@upy	1312, 1317, 1324, 1327, 1330, 1333, 1338, 1349, 1354, 1362, 1371
\@MXV@upyprime	1354, 1362
\@MXV@upz	1313, 1318, 1325, 1328, 1331, 1334, 1335, 1351, 1356, 1363, 1371
\@MXV@upzprime	1356, 1363
\@MXV@varray	778, 779, 1283, 1593, 1620
\@MXV@viewobj	1591, 1629, 1664, 1669
\@MXV@viewscount	1266, 1581, 1583, 1619, 1671, 1673, 1675, 1677, 1680, 1692, 1697, 1699, 1708, 1714, 1716, 1724

\@MXV@viewsfileii
 .. 1289, 1409, 1434, 1442, 1449, 1456, 1463,
 1470, 1477, 1489, 1498, 1504, 1520, 1526, 1537,
 1545, 1553, 1561, 1569, 1612, 1634, 1638, 1787
 \@MXV@viewsprovidedtrue 1594
 \@MXV@viewx 1294,
 1297, 1301, 1320, 1323, 1335, 1338, 1366, 1372
 \@MXV@viewy 1295,
 1298, 1302, 1320, 1324, 1334, 1337, 1367, 1372
 \@MXV@viewz 1296, 1299,
 1303, 1309, 1323–1325, 1333, 1336, 1368, 1372
 \@MXV@warning 216, 217
 \@MXV@xname . 1379, 1413, 1582, 1583, 1627, 1682, 1683
 \@auxout 207
 \@break@tfor 699
 \@bsphack 207
 \@onlypreamble 150, 170, 171, 1111
 \@rgi 90, 91
 \@tstii 90, 91
 \@x 97, 102–104, 109–111, 130–132, 137–139, 143–145
 3DOptions (3D) (key) 40
 3DResources (3D) (key) 40

_ 97

A

\AcrobatVer 96, 98, 105,
 113, 119, 121, 124, 127, 128, 130, 134, 135, 143
 \AcroVer 5, 19, 20, 94, 150, 151
 action (cuepoints) (key) 22
 \additional@Instances 655, 795, 1284
 \adj@measure 1081, 1086, 1091, 1099
 \aeb@argi 272, 281, 282
 \aeb@argii 272, 281
 \aeb@argiii 273, 282
 all (key) 11
 \and 1658, 1663
 \appType 88, 90
 \arg 526, 528
 \AtBeginDocument 297
 \AtEndDocument 218
 \audCtrlHt 9, 234
 \audCtrlWd 9, 233
 AudioPlayer (key) 11

B

background (3D) (key) 42
 \Bbox 752
 \bboxOf 236, 1107

\BC 55
 \BG 55
 \boolean 1642, 1657, 1658, 1662, 1663
 borderwidth (key) 15

C

\CA 55
 \calc@prop 726, 727, 741, 742, 1086
 \cntrlbrHt 234
 \cntrlbrWd 233
 cuepoints (key) 21

D

deactivated (key) 13
 \DeclareOptionX 17, 64
 \defaultPoster 1076
 \define@boolkey 320, 367, 381, 385, 396, 401, 406, 440
 \define@choicekey 87,
 304, 312, 336, 343, 369, 411, 554, 1795, 1803
 \defineRMPPath 153
 \dimen@ ... 31, 32, 39, 42–44, 47, 50–52, 601, 602,
 624, 626, 627, 629, 632, 635–637, 640, 643–645

E

\egroup 33, 54, 603, 647
 \embedEPS 230, 1106
 enabled (key) 13
 \endinput 7, 10, 15, 63
 \endlinechar 1636
 \equal 212, 214, 1380, 1385,
 1390, 1395, 1400, 1502, 1524, 1582, 1585, 1588

F

\Ff 56
 \FfReadOnly 56
 \filename@area 259, 481, 610, 1164,
 1169, 1190, 1195, 1217, 1222, 1245, 1250, 1752
 \filename@base 257, 259, 481, 611, 1164,
 1169, 1191, 1196, 1218, 1223, 1246, 1251, 1752
 \filename@ext ... 255, 257, 259, 481, 612, 1164,
 1169, 1191, 1196, 1218, 1223, 1246, 1251, 1752
 \filename@parse
 ... 251, 471, 605, 1152, 1176, 1203, 1230, 1737
 \FileStrmAudioPlayer 709
 \FileStrmVideoPlayer 704
 flashvars (key) 21
 foreground (3D) (key) 41
 \FPifeq 1321, 1365
 \FPifneg 1309
 \FPpi 1304, 1305

\FPupn 1294–1297, 1301–1308,
1311–1313, 1316–1318, 1320, 1323–1325, 1327,
1329–1331, 1333, 1335, 1337, 1340, 1342–1344,
1346, 1348, 1350, 1352, 1354, 1356, 1358–1368

G

\getargsiii 272, 280
\Gin@defaultbp 723, 724, 738, 739

H

height (key) 16
\Hy@unicodedefalse 82
\hyper@normalise 154, 246

I

\if@FndSuppExt 593, 702
\if@MXV@defaultviewprovided 1628
\ifdim 725, 740
\ifeof 1640, 1648
\IfFileExists 1634, 1786
\ifKV@rmAnnot@defaultposter 722
\ifKV@rmAnnot@modeltree 402
\ifKV@rmAnnot@passcontext 407
\ifKV@rmAnnot@skinAutoHide 441
\ifKV@rmAnnot@toolbar 397
\ifKV@rmAnnot@transparentBG 386
\ifKV@rmAnnot@url 817, 823, 884, 904, 988, 998
\ifKV@rmAnnot@windowed 956, 963
\ifpdf 6
\ifrma@EmbedAudioPlayer 78, 584, 1061
\ifrma@EmbedFile 75, 999
\ifrma@EmbedVideoPlayer 76, 576, 1043
\ifrma@isiiid 459, 461, 649, 674, 776,
792, 811, 841, 857, 869, 876, 882, 916, 943, 1011
\ifthenelse 211, 216, 1380, 1385, 1390, 1395,
1400, 1502, 1524, 1582, 1585, 1588, 1657, 1662
\ifuseWinAcrobat 86, 99, 106, 118
\ifVideoPlayerEx 156, 826, 1013
\ifxetex 9
\InputIfFileExists 65, 228
invisible (key) 16
\isundefined 216

J

\JS 547

K

keys:
3DOptions (3D) 40
3DResources (3D) 40

action (cuepoints) 22
all 11
AudioPlayer 11
background (3D) 42
borderwidth 15
cuepoints 21
deactivated 13
enabled 13
flashvars 21
foreground (3D) 41
height 16
invisible 16
material (3D) 43
modeltree 16
name 13
name (cuepoints) 22
noChange 12
none 12
none (3D) 41
passcontext 17
playcount 18
poster 15
posternote 15
resources 19
scale 16
skin 17
skin1 11
skin2 11
skin3 11
skin4 11
skin5 11
skin6 11
skin7 11
skinAutoHide 18
skinBGAlpha 18
skinBGColor 18
speed 18
time (cuepoints) 22
toolbar 16
transparentBG 16
type (cuepoints) 22
url 15
VideoPlayer 11
volume 18
width 16
windowed 13

L

\literalps@out 659, 755, 1669, 1695, 1712

M		\PackageWarning 159, 310, 318, 376, 427, 430, 559, 1525, 1800, 1810
\makeJSSpecials 265, 598	\PackageWarningNoLine 58, 219	
\makePoster 23, 1105	passcontext (key) 17	
material (3D) (key) 43	\pathToPlayers 81	
\mmGetMetaData 187	\PathToSkins 83, 285–288, 290–295	
\mmGetSource 185	\pathToSkins 81	
\mmGetVersion 190	\pdf@rect 752	
\mmGetVideoState 188	\pdfstringdef 82, 152, 1379, 1413, 1511, 1583, 1627	
\mmIsLooping 192	playcount (key) 18	
\mmMute 178	\PMPV 595	
\mmNextCuePoint 175	poster (key) 15	
\mmPause 173	posternote (key) 15	
\mmPlay 172	\previewOn 55	
\mmPrevCuePoint 176	\ProcessOptionsX 18, 70	
\mmRewind 174	\protected@write 207	
\mmSeek 177	\providecommand 354, 362, 1105	
\mmSeekCuePoint 183	\ps@mark 195, 238, 383, 495, 496, 499, 500, 507–509, 515, 534, 537, 660, 756, 773, 774, 784, 785, 791, 793, 798, 801, 804, 807, 808, 810, 812, 813, 818, 819, 824, 825, 829, 830, 833, 834, 836–839, 842, 843, 846, 847, 849, 850, 852, 853, 873, 874, 909, 910, 939, 941, 964, 965, 986, 987, 1000, 1002, 1004, 1008, 1015, 1017, 1024, 1026, 1029, 1033, 1036, 1037, 1044, 1045, 1047, 1051, 1054, 1055, 1062, 1063, 1065, 1069, 1108, 1155, 1158, 1160, 1179, 1182, 1184, 1206, 1209, 1211, 1233, 1236, 1238, 1671, 1673, 1675, 1677, 1696, 1699, 1713, 1716, 1741	
\mmSetScaleMode 189	\pushButton 55	
\mmSetStageColor 191		
\mmShowLoopButton 194	R	
\mmSkin 181	\ratio 40, 48, 633, 641	
\mmSkinAlpha 184	\read 1639, 1647	
\mmSkinAutoHide 193	\real 625, 628	
\mmSkinColor 182	\RefObjRm 9, 227	
\mmSource 180	\RequirePackage 2–4, 16, 66, 71, 73	
\mmUseLocal 186	\resetWindowDimPos 362	
\mmVolume 179	resources (key) 19	
modeltree (key) 16	\rm@@temp@@exp 22, 196	
N		\RM@action 7, 10, 12, 15, 63
\Name 266, 596	\rm@Annot@height 32, 41, 44, 49, 50, 56, 602, 625, 626, 634, 637, 642, 643, 724, 725, 739, 740, 752	
name (key) 13	\rm@Annot@width 31, 41, 42, 49, 52, 56, 601, 628, 629, 634, 635, 642, 645, 723, 725, 738, 740, 752	
name (cuepoints) (key) 22	\rm@appendFileSpecs 20, 489, 498, 1163, 1190, 1217, 1245, 1760	
\newboolean 1265, 1267, 1268	\rm@appendNameTree 20, 486, 494, 1168, 1195, 1222, 1250, 1757	
noChange (key) 12	\rm@argii 243, 250, 256, 258, 1121	
none (key) 12		
none (3D) (key) 41		
\not 214, 1642, 1657, 1662, 1663		
O		
\openin 1638		
options:		
use3D 4		
P		
\p@thHash 91, 93, 100, 107		
\PackageError 114, 1113, 1120, 1126, 1408, 1433, 1441, 1448, 1455, 1462, 1469, 1476, 1488, 1497, 1503, 1519, 1536, 1544, 1552, 1560, 1568, 1611, 1789		

\rm@csarg	80, 198,	\rma@isVPEEmbedded	77, 577, 579
200, 249, 250, 256, 258, 281, 282, 368, 433,		\rma@mimeType	679, 683, 688, 697, 707, 1006
438, 476, 483, 607, 1023, 1073, 1144, 1747, 1754		\rma@mimeType@mpiii	707, 1067
\rm@ctrlName	153, 155	\rma@mimeType@prc	683
\rm@defineURLPath	154, 155	\rma@mimeType@swf	688, 1031, 1049
\rm@irfstrm	1147,	\rma@mimeType@uiiid	679
1165, 1170, 1192, 1197, 1219, 1224, 1247, 1252		\rma@next	461, 462
\rm@One	5, 433, 438, 577, 585, 1023	\rma@nResources	458,
\rm@saveNamedPath	246, 248	460, 469, 477, 480, 524, 527, 531, 535, 537, 540	
\rm@SkinsAndPlayerPaths	284, 297	\rma@pathToPlayers	84, 152, 1045, 1063
\rm@thisMimeType	245, 250	\rma@pathToSkins	82–84, 1027
\rm@thisPath	244, 249	\rma@PkEr@i	713, 1112
\rm@Zero	5, 77, 79, 579, 587, 1073	\rma@PkEr@ii	244, 1119
\rma@addFileSpecs		\rma@PkEr@iii	252, 473, 1125, 1739
467, 490, 599, 663, 1010, 1011, 1013, 1733, 1761		\rma@poster@descrip	379, 675, 686, 691, 708
\rma@addResources		\rma@posternote	380, 1093
... 466, 487, 599, 656, 814, 820, 826, 1732, 1758		\rma@proc@resources	462, 464
\rma@Annot@name	303, 650, 654, 753, 754, 759	\rma@ps@bg@setcolor	1079, 1087, 1094
\rma@ANT	302, 495, 496, 809, 810, 812, 813,	\rma@ps@font	1082, 1090, 1103
818, 819, 824, 825, 829, 830, 833, 834, 836–839		\rma@ps@fontsize	1083, 1092, 1100, 1101
\rma@appendToNameTree	652, 656, 1167, 1194, 1221, 1249	\rma@ps@msg	1084, 1093
\rma@arg	436, 438, 468, 471, 473, 474, 483,	\rma@ps@relfontsize	1083, 1091, 1099, 1102
1735, 1737, 1739, 1742, 1745, 1748, 1751, 1754		\rma@ps@txt@setcolor	1082, 1089, 1097
\rma@array@hold	525, 530, 897	\rma@ps@txt@x	1081, 1088, 1095
\rma@basefilename	611, 715	\rma@ps@txt@y	1081, 1088, 1096
\rma@ckFileForEmbed ...	1141, 1153, 1177, 1204, 1231	\rma@psgraphics@poster	731, 746, 1076
\rma@data	274, 280	\rma@recordAudioPlayer	583, 711
\rma@dict@hold	525, 533, 938	\rma@recordVideoPlayer	575, 699
\rma@edefexecute		\rma@requirefp	66, 69, 72
..... 22, 34, 196, 251, 253, 470, 528, 605,		\rma@resource	263, 266, 268, 269, 596
613, 619, 653, 672, 1151, 1175, 1202, 1229, 1736		\rma@rmAnnot@borderwidth	372–375, 765, 767
\rma@embed	478, 482, 1145, 1146, 1165, 1170,	\rma@rmAnnot@cuepoints	522, 523, 526, 896, 938
1191, 1196, 1218, 1223, 1246, 1251, 1749, 1753		\rma@rmAnnot@deactivated	315–317, 934
\rma@embed@mpiii@Poster	229, 231, 260	\rma@rmAnnot@enabled	307–309, 914
\rma@EmbedAudioPlayerfalse	78, 585	\rma@rmAnnot@flashvars	519, 718, 720, 900, 901
\rma@EmbedAudioPlayertrue	588	\rma@rmAnnot@iiiDjs ...	657, 1282, 1731, 1732, 1735
\rma@EmbedFilefalse	609	\rma@rmAnnot@modeltree	403, 404, 944
\rma@EmbedFiletrue	75, 608	\rma@rmAnnot@PassContextClick	408, 409, 950
\rma@EmbedVideoPlayerfalse	76, 577	\rma@rmAnnot@playcount	455, 922
\rma@EmbedVideoPlayertrue	580	\rma@rmAnnot@poster	
\rma@extension	612, 614, 673, 715 378, 589, 721, 733, 736, 748, 760, 761	
\rma@fs@expand	474–	\rma@rmAnnot@resources	465, 466, 468, 710, 877
476, 482, 606, 607, 1142–1144, 1745–1747, 1753		\rma@rmAnnot@Skin	415, 417–424, 432,
\rma@fullpath	716, 885, 904, 989, 1002	828, 829, 889, 890, 1014, 1018, 1020, 1027, 1072	
\rma@input@iiidCode	65, 68, 648, 1818	\rma@rmAnnot@speed	452, 921
\rma@Instances	652, 655, 1154, 1178, 1205, 1232	\rma@rmAnnot@toolbar	398, 399, 945
\rma@invisible	237, 383, 384, 730, 745	\rma@rmAnnot@transparent	387, 388, 946, 949
\rma@isAPEEmbedded	79, 585, 587	\rma@rmAnnot@type .	678, 682, 688, 691, 707, 717,
\rma@isiidfalse	459, 617	719, 735, 797, 800, 816, 822, 845, 848, 862, 864,	
\rma@isiidtrue	615, 616	883, 899, 937, 948, 1010, 1012, 1035, 1053, 1071	

\rma@rmAnnot@type@flv 717,
 797, 822, 845, 862, 883, 937, 1012, 1035, 1071
 \rma@rmAnnot@type@mpiii .. 260, 706, 719, 735, 1053
 \rma@rmAnnot@type@prc 616, 681
 \rma@rmAnnot@type@swf
 687, 800, 816, 848, 864, 899, 948, 1010
 \rma@rmAnnot@type@uiiid 615, 677
 \rma@rmAnnot@volume 450, 895, 906
 \rma@RMCSUBTYPE 676, 689, 703, 708, 788
 \rma@rmCuePt@action 546, 547, 568
 \rma@rmCuePt@name 539, 542, 562
 \rma@rmCuePt@time 545, 565
 \rma@rmCuePt@type 544, 557, 558
 \rma@set@mpiiiposter 235, 240, 589, 590
 \rma@skinAutoHide 442, 443, 892
 \rma@skinBGAlpha 448, 894
 \rma@skinBGColor 446, 893
 \rma@skinName 413, 416, 431, 830, 1015,
 1017, 1021, 1023, 1024, 1026, 1029, 1033, 1073
 \rma@tempi
 . 254, 260, 614–616, 673, 677, 681, 687, 694, 706
 \rma@thisfileName
 . 715, 716, 812, 818, 824, 838, 887, 905, 992, 993
 \rma@thisfilepath 610, 716
 \rma@tmp@exp 479, 484, 1750, 1755
 \rma@toks 457, 460,
 479, 484, 488, 491, 1734, 1750, 1755, 1759, 1762
 \rma@type 693, 695, 698
 \rma@urlresource 264, 266, 270, 271, 597
 \rma@useNamedPath 229,
 471, 473, 605, 1152, 1176, 1203, 1230, 1737, 1739
 \rma@VideoPlayer 157, 167, 285, 833, 1039, 1040, 1045
 \rma@winDimPos@height 328, 358
 \rma@winDimPos@position 335, 359
 \rma@winDimPos@width 321, 357
 \rma@winDimPosHeight@def 330, 968
 \rma@winDimPosHeight@max 332, 969
 \rma@winDimPosHeight@min 334, 970
 \rma@winDimPosPos@halign 339–341, 979
 \rma@winDimPosPos@hoffset 351, 981
 \rma@winDimPosPos@valign 346–348, 980
 \rma@winDimPosPos@voffset 353, 982
 \rma@winDimPosWidth@def 323, 973
 \rma@winDimPosWidth@max 325, 974
 \rma@winDimPosWidth@min 327, 975
 \rmaName 268
 \rmaNameP 269
 \rmAnnot 29, 594
 \rmAnnot@height
 27, 28, 37, 40, 43, 392, 393, 622, 633, 636

\rmAnnot@scale 25, 394, 395, 623, 625, 628
 \rmAnnot@width 24, 26, 36, 48, 51, 390, 391, 621, 641, 644
 \rmArgi 95, 96
 \rmaUrlName 268
 \rmaUrlNameP 271
 \rmBeta 95, 105
 \rmDC 95, 98
 \rmiiid@addToScriptsArray 662, 1734, 1740
 \rmiiiDOpts@bg@flashvars 1214, 1259
 \rmiiiDOpts@bg@rName
 . 1203, 1204, 1207, 1210–1212, 1217, 1222, 1258
 \rmiiiDOpts@fg@flashvars 1187, 1257
 \rmiiiDOpts@fg@rName
 . 1176, 1177, 1180, 1183–1185, 1190, 1195, 1256
 \rmiiiDOpts@mat@flashvars 1242, 1263
 \rmiiiDOpts@mat@mName 1241, 1261
 \rmiiiDOpts@mat@rName
 . 1230, 1231, 1234, 1237–1239, 1245, 1250, 1260
 \rmiiiDOpts@no@rName
 . 1152, 1153, 1156, 1159–1161, 1163, 1168, 1255
 \rmiiiDOpts@TLRes 1138
 \rmiiiDTLOpts 1134
 \rmSkinPath
 . 100, 102, 107, 109, 120, 123, 129, 136, 142, 148
 \romanVer 134, 135, 137

S

\S 56
 \saveNamedPath 10, 21, 229, 285–289, 291–295
 scale (key) 16
 \setboolean 1290, 1291, 1640,
 1641, 1649, 1651, 1767, 1771, 1775, 1779, 1783
 \setPosterProps 38, 1104
 \setRmOptions 57, 368
 \setRmOptions3D 368
 \setWindowDimPos 354, 363
 skin (key) 17
 skin1 (key) 11
 skin2 (key) 11
 skin3 (key) 11
 skin4 (key) 11
 skin5 (key) 11
 skin6 (key) 11
 skin7 (key) 11
 skinAutoHide (key) 18
 skinBGAlpha (key) 18
 skinBGColor (key) 18
 speed (key) 18
 \stepcounter 604

T		U	
\temp@expand@sets	356, 360	\uppercase	254, 613, 672
\therm@Cnt	302, 303, 477, 496, 499, 500, 531, 535, 537, 608, 660, 666, 704, 709, 729, 733, 744, 748, 753, 754, 756, 770, 771, 773, 774, 781, 782, 784, 785, 787, 789, 791, 793, 798, 801, 804, 807, 808, 813, 819, 825, 839, 842, 843, 846, 847, 849, 850, 852, 853, 859, 865, 870, 873, 874, 909, 910, 915, 930, 939, 941, 958, 964, 965, 986, 987, 1145, 1155, 1156, 1158, 1160, 1161, 1179, 1180, 1182, 1184, 1185, 1206, 1207, 1209, 1211, 1212, 1233, 1234, 1236, 1238, 1239, 1671, 1673, 1675, 1677, 1680, 1692, 1697, 1699, 1708, 1714, 1716, 1724, 1741, 1742, 1748	url (key)	15
\this@bbox	728, 729, 743, 744	\urlName	266, 597
\this@height	724, 727, 728, 739, 742, 743, 1080	use3D (option)	4
\this@width	723, 726, 728, 738, 741, 743, 1080	\useVideoPlayerPlus	156, 168, 170
time (cuepoints) (key)	22	\useVideoPlayerX	164, 166
toolbar (key)	16	\useVideoPlayX	161, 163
transparentBG (key)	16	\useWinAcrobatfalse	89
type (cuepoints) (key)	22	\useWinAcrobattrue	86, 88
		V	
		VideoPlayer (key)	11
		\VideoPlayerExfalse	156
		\VideoPlayerExtrue	166
		volume (key)	18
		W	
		\whiledo	1642
		width (key)	16
		windowed (key)	13

6 Change History

v1.0 (2010/09/24)	General: Made a correction to a typo for CuePoints (I had Cuepoints). Now the cue points feature works. Removed some spurious spaces from \rmAnnot	Added support for \Name and \urlName inside a eform field.	2
v1.0a (2010/09/29)	General: Added an invisible option for \rmAnnot. The annot is given a transparent poster, if no poster is specified.	v1.1a (2010/17/07)	General: Added definitions for none and noChange that can be recognized by JS to indicate that the video should have no skin, and the video should have the same skin as the previous video (noChange)
v1.0b (2010/09/29)	General: Experimental use of VideoPlayerPlus.swf and VideoPlayerX.swf	v1.1b (2010/14/12)	General: Corrected the definition of borderwidth. Acrobat uses for none, thin, medium, and thick the values of 0,1,3,5 not 0,1,2,3 (for some reason).
v1.0c (2010/10/01)	General: Added \let\rma@addResources\@empty \let\rma@addFileSpecs\@empty to clean out these macros, if one \rmAnnot had resources, these were included in the next \rmAnnot as well causing distiller to cough a fur ball. The problem manifests itself when \useVideoPlayerPlus or \useVideoPlayerX is used.	v1.1d (2010/27/12)	General: Made some minor changes, labeled as 12/27/10 (2010/22/12 v1.1c) Made the definition \saveNamedPath global, needed for the fldigital package.
v1.1 (2010/10/07)	General: Added definitions for skins so they can be referenced by name in the resources key.	v2.0.6 (2018/03/21)	\pathToPlayers: Placed \pathToSkins in a group and disallowed unicode encoding by hyperref
		v2.0a (2011/09/11)	\pathToPlayers: Arguments now pass through

\pdfstringdef	4	poster	39
v2.0b (2015/09/30)		Added all skin name	17
\pathToPlayers: Added \AcroVer and a more		Added defaultposter designed for use with	
intelligent method of finding the path to		MP3s	16
skins and players.	5	Added an all skin	11
v2.0d (2016/10/09)		\pathToPlayers: Added 32 and 64 as values of	
General: Added 10.4 function		the win key of \AcroVer	5
\mmShowLoopButton	8	\rmAnnot: Added pmpv to \rmAnnot, now	
\useVideoPlayerPlus: Removed support for		displays name of the file to play	24
VideoPlayerPlus.swf.	7	Support for keys width and height of the	
v2.2 (2020/08/21)		rmAnnot family. The scale key is also	
General: Add user interface to design of default		defined.	25