

# The `ocgtools` package<sup>\*†</sup>

Robert Mařík  
marik@mendelu.cz

March 31, 2010

## 1 Introduction

The package `ocgtools` is designed to insert OGC (Optional Group Content) into PDF presentations. From the user's point of view, the package allows to insert in a comfortable way any  $\text{\TeX}$  material into separate layer in PDF document and insert links which toggle this layer on/off. Hence parts of PDF document like formatted text, tables, math formulas or graphics can be turned to visible or invisible state by clicking active links or buttons.

Similar packages are `cooltooltips`, `pdfcomment`, `AcroTeX` and `fancytooltips`. In contrast to `cooltooltips` and `pdfcomment`, we can work with any  $\text{\TeX}$  material, not only plain text. In contrast to `fancytooltips`, no external file is necessary and more minilayers (see below) can be opened simultaneously on one page. However, `fancytooltips` and `ocgtools` can be combined in the same document, even on the same page. `AcroTeX` has far more possibilities than `ocgtools`, but has three disadvantages: (1) limited support for `pdftex` (no layers with `pdftex` driver) (2) PDF file needs postprocessing (3) for the post-processing the non-free Adobe Acrobat Professional is necessary. However, the user of `ocgtools` must have `AcroTeX` installed, since we use its capability to insert buttons and JavaScripts into document (`eforms` and `insdljs` packages).

Two types of OCG objects can be inserted

- *layers*: OCG's which span across the whole size of paper (scaled if necessary)
- *minilayers*: OCG's which have their natural size and are placed somewhere close to the link which toggles them on/off.

The user should use viewer which allows to hide/reveal layers by JavaScript. This includes especially Adobe Reader. Some limited functionality is also in Foxit Reader (see the option `nobutton` below). In some other viewers (like `xpdf`) users see red warning on the first page (see `\ocgtools@msg`), all layers are visible and cannot be hidden.

**Big warning:** Unfortunately, with this package you may make your PDF files **less portable** even if you use Adobe Reader! Till now, we observed the following problems:

- `.jpg` pictures may look darker when using `transparent` option and **Linux** version of Adobe Reader – Adobe Reader switches to different rendering method which seems to be system dependent. Hope, Adobe fixes this problem soon.
- The layer with initial message on the first page which should be hidden to the users of Adobe Reader is sometimes still visible, if the PDF file is opened in Internet browser. From this reason, layer `ocgtools0` is switched on and then off when initializing PDF document. Hope, this workaround solves the problem. If not, report the problem with a minimal example, please.

---

<sup>\*</sup>This document corresponds to `ocgtools` v0.5, dated 2010/04/01.

<sup>†</sup>Supported by grant 131/2010 of Higher Education Development Fund (FRVŠ)

Important  
comment!

## 2 Examples

Several examples are distributed with the package. We have one minimal example, one example which shows cooperation with `preview.sty` and mainly – examples which show cooperation with three most popular packages to build presentations. This includes `web.sty`, `pdfscreen.sty` and `beamer.cls`. We have three demo files for each – with no panel, with panel on the right and with panel on the left. All examples can be recompiled by running `ocgtools-test.bat` on Windows and `ocgtools-test.sh` on Linux.

## 3 Usage

All the packages we use are on CTAN or in TeXlive2009. The file `ocg.sty` is a part of `asymptote` package<sup>1</sup>. AcroTeX is quite old on CTAN but the version on CTAN works well. New version of AcroTeX is at [www.acrotex.net](http://www.acrotex.net), if necessary.

**Prerequisites:** Only `pdflatex` is supported. The route via `dvips` or `dvipdfm` is not (yet?) supported (this restriction follows from `ocg.sty` and `transparent.sty` packages). To work with the package load `color.sty` and `hyperref.sty` packages. (This is usually done automatically by most packages which are used to build PDF presentations. If not, these packages are loaded by `ocgtools` automatically as soon as `\definecolor` and `\href` remain undefined, respectively.) You may use also `xcolor.sty`, but this package is not compatible with `transparent` option.

You have to compile your `.tex` file *three times* (!). If you change your document and create new layer, you have to compile three times again. After most changes (which include change in position of but not the number of layers) you have to compile twice to put layers on correct position. If you change the content of text layers or minilayers, one compilation is sufficient.

Load package `ocgtools.sty` as usual: `\usepackage[options]{ocgtools}`. Options include:

**transparent** Layers in fact do not span over the whole page, but the material of layer is scaled to some reasonable size (we keep aspect ratio). If the layer is activated, the page is covered by a uniform color `ocgbg` and the layer is placed on the top. The options `transparent` makes the color `ocgbg` transparent. Looks cool, but it could be slow and could change some colors of bitmap pictures (and this behavior seems to be system dependent – looks differently on Linux and differently on Windows). The transparency is achieved by putting the command `\ocgtools@transparent` on appropriate place. The default setting is `\def\ocgtools@transparent{\transparent{0.4}}`. Use something like the following three lines to override:

```
\makeatletter
\def\ocgtools@transparent{\transparent{0.6}}
\makeatother
```

**nobutton** The pages with active layers have a transparent button which can be used to hide this layer. The user simply clicks anywhere and the layers become hidden. In some viewers (like Foxit Reader) the button is not 100% transparent. This option allows not to include the big button to hide layers. The layers can be closed by clicking the red cross below.

**insertvisible** In some rare cases problems with OCG's occur on Windows installation of TeX. Temporary (I hope) workaround is to use this option, which inserts OCG's as visible. The visibility is turned off when the PDF is opened on the first page.

---

<sup>1</sup>Do not interchange with the `ocg.sty` provided by Österreichische Computer Gesellschaft. The correct file `ocg.sty` is at <http://www.tug.org/svn/texlive/trunk/Master/texmf/tex/latex/asymptote/ocg.sty>

**mouseover** Layers can be opened by MouseOver action. The corresponding active area is an invisible square with side 8pt placed on the right bottom corner of the text or picture which opens the layer. More details: MouseOver opens the layer and MouseExit hides this layer again. If mouse button is clicked, the layer remains opened after exiting the button.

**minimouseover** As mouseover, but restricted to minilayers.

**noocg** The package and its commands are ignored.

**noprogressmsg** Opening PDF on the first page show the message related to initial processing layers. This option turns the message off.

**nopageclose** By default, all layers become off and all buttons become hidden if the page is changed. The option **nopageclose** suppresses this behavior.

**\ocgtext** The macro **\ocgtext**[ $\langle width \rangle$ ]{ $\langle text1 \rangle$ }{ $\langle text2 \rangle$ } is used to create layer which contains any L<sup>A</sup>T<sub>E</sub>X material.  $\langle text1 \rangle$  is a text which is written in an “usual” way (in blue color which indicates that this text can be used to hide/unhide another object) and this text is used to hide/unhide layer with  $\langle text2 \rangle$  (which could be text divided into more than one paragraph, figure created by **mfpic** or any L<sup>A</sup>T<sub>E</sub>X material).  $\langle text2 \rangle$  is either placed into **\hbox** (if  $\langle width \rangle$  is 0pt, which is default) or wrapped by **\vbox** with **\hsize** equal to  $\langle width \rangle$  (in the opposite case).

**\ocgpicture** The macro **\ocgpicture**[ $\langle params \rangle$ ]{ $\langle picture \rangle$ } is used to insert a picture which is used as a link to layer with bigger version of this picture. The optional parameter  $\langle params \rangle$  is used by **\includegraphics** command to draw picture in text. The layer contains scaled version of the picture (aspect ratio is preserved).

**\ocgminitext** The macro **\ocgminitext**[ $\langle width \rangle$ ]{ $\langle text1 \rangle$ }{ $\langle text2 \rangle$ } is used to create layer containing  $\langle text2 \rangle$  which is placed near the right top corner of  $\langle text1 \rangle$ .  $\langle text2 \rangle$  is placed into **\hbox** or **\vbox** with given **hsize** according to the value of  $\langle width \rangle$ , as has been explained at **\ocgtext** macro. Macros **\ocgminitextrb**, **\ocgminitextlt** and **\ocgminitextlb** can be used to place the minilayer to the right bottom, left to and left bottom corner.

**\ocgtextstart** Macros **\ocgtextstart** and **\ocgtextend** are used to denote the start and the end of hyperlink which is used to hide/reveal layers. Default setting is **\def\ocgtextstart{\color{blue}}** and **\def\ocgtextend{}** and hence, the links are blue and there is no mark at the end of the link.

**ocgpaper** The colors **ocgbg** and **ocgpaper** are used to set the colors which are used to hide page and color which is used as background for text produced by **\ocgtext** and **\ocgminitext** commands. The default setting is **\definecolor{ocgpaper}{rgb}{1,1,0.2}** and **\definecolor{ocgbg}{rgb}{0,0,0}**.

**\layerHshift** The dimensions **\layerHshift** and **\layerVshift** are used to place layers exactly on the top of the page. If presentation packages **web**, **beamer** or **pdfscreen** are used, then the length is set automatically.

**\ocgtools@shipoutstart@hook** **\ocgtools@shipoutstart@hook** and **\ocgtools@shipoutend@hook** are introduced to insert some material at the begin and at the end of the box with the page contents. Can be used for example to put background to the presentation. Both commands should produce boxes of zero dimensions, i.e. use something like

```
\makeatletter \def\ocgtools@shipoutstart@hook{\hbox to 0 pt{\kern
-1in \Huge A \hss}} \makeatother
```

to insert letter “A” into the left bottom corner. Note that you will see this letter only if the background of the presentation is transparent. See also the files **ocgtools-example-web\*.tex** and **ocgtools-example-web\*.pdf** for slightly more complicated background.

**\ocgclosechar** Macro **\ocgclosechar** is used to print mark which is used to close layers and minilayers and **\ocgtools@msg** **\ocgtools@msg** contains string for users of viewers which do not support layers (see the first few lines of the code for default settings).

## 4 Possible future development and known problems

1. Test and support more packages for slides than current `web.sty`, `pdfscreen.sty` and `beamer.sty`? Which ones?
2. Using `preview` package it is possible to extract equations and figures from the document and redefine `\ref` and `\eqref` commands in such a way that clicking (or mouseover) opens on the current page the layer with this equation (figure) and shift clicking moves the user to the page with this equation (figure). Or would `fancytooltips` produce better results in this case (smaller PDF file)? See also the paper by Ross More at <http://www.tug.org/TUGboat/Articles/tb29-3/tb93moore.pdf>. See the `ocgtools-preview.*` files for some initial attempts in this direction.
3. Introduce draft mode, which prints all layers at the end of document with links there and back?
4. Improve placement of minitextlayers. Currently we have one the reference point for both minitextlayers aligned to top and bottom, which is far from optimal.
5. In some rare cases the layers may be not inserted properly when the  $\text{\LaTeX}$  file is compiled on **Windows** – see the option `insertvisible` which solves this problems in most cases. Since the author has limited access to  $\text{\TeX}$  installations on Windows and hence reporting of problems with a minimal example is highly appreciated.
6. The package cannot be used to hide 3D graphics inserted by `movie15` package. However, you can put this graphics into floating window or full-screen using capabilities of Acrobat Reader 9.
7. If you open and close layer and then use "Back" (Alt+LeftArrow), the layer opens again, but the button to hide this layer becomes unavailable. Possible solution is to define open action and close action for each layer (is it possible in current PDF specification?).

## 5 Implementation

Initial settings

```
1 (*package)
2 \def\ocgtools@msg{If message does not disappear after a short time, the
3   author either did not compile the \LaTeX{} file three times, or your
4   PDF viewer does not support OCG. Use Adobe Reader!}
5
6 \def\ocgtextstart{\color{blue}}
7 \def\ocgtextend{}
8 \ifx\definecolor\undefined\RequirePackage{color}\fi
9 \ifx\href\undefined\RequirePackage[pdftex]{hyperref}\fi
10 \definecolor{ocgpaper}{rgb}{1,1,0.2}
11 \definecolor{ocgbg}{rgb}{0,0,0}
12 \RequirePackage{graphicx}
13 \RequirePackage{pifont}
14 \RequirePackage{ocg}
15
16 \def\ocgclosechar{\ding{56}}
```

Make packages and options known.

```
17 \RequirePackage{atbegshi}
18 \RequirePackage[pdftex]{eforms}
19 \newif\if@ocgtools@transparent\@ocgtools@transparentfalse
20 \DeclareOption{transparent}{\@ocgtools@transparenttrue}
```

```

21 \newif\if@ocgtools@insertvisible\@ocgtools@insertvisiblefalse
22 \def\ocgtools@initialvisibility{0}
23 \DeclareOption{insertvisible}{\@ocgtools@insertvisibletrue
24   \def\ocgtools@initialvisibility{1}}
25 \newif\ifocg@hide@button\ocg@hide@buttontrue
26 \DeclareOption{nobutton}{\ocg@hide@buttonfalse}
27 \newif\ifocgtools@noocg\ocgtools@noocgfalse
28 \DeclareOption{noocg}{\ocgtools@noocgtrue}
29 \def\ocgtools@progressmsg{\lower \layerVshift\hbox to 0 pt{ %space
30   \textField[\V{OCGtools: processing OCG's ...}\BG{1}
31   \textColor{1 0 0} \textSize{10}}}%
32 {ocgtoolsmsg}{6cm}{20pt}\hss}}%
33 \DeclareOption{noprogressmsg}{\let\ocgtools@progressmsg\relax}
34 \newif\if@ocgtools@minimouseover \@ocgtools@minimouseoverfalse
35 \DeclareOption{minimouseover}{\@ocgtools@minimouseovertrue}
36 \newif\if@ocgtools@mouseover \@ocgtools@mouseoverfalse
37 \DeclareOption{mouseover}{\@ocgtools@mouseovertrue \@ocgtools@minimouseovertrue}
38 \newif\if@ocgtools@pageclose \@ocgtools@pageclosetrue
39 \DeclareOption{nopageclose}{\@ocgtools@pageclosefalse}
40 \ProcessOptions

```

We insert JavaScripts which are evaluated when the file is opened at the first page.

```

41 \edef\ocgtools@pdfpageattr{
42 /AA << /O << /S /JavaScript /JS
43 (

```

We close layers, if the page is opened. If the document is opened and `ocgtoolsOCGs` is not initialized, we skip to catch part and initialize.

```

44   try{
45     var temp = ocgtoolsOCGs.length;
46     \if@ocgtools@pageclose
47     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
48       {
49         ocgtoolsOCGs[i].state = false;
50       }
51     \fi
52   }

```

We initialize document – we find all layers, put into `ocgtoolsOCGs` field and make them hidden.

```

53   catch (e){

```

No dotted rectangle for buttons which have focus.

```

54     app.focusRectg = false;

```

The OCG objects inserted by `ocgtools` are stored in variable `ocgtoolsOCGs` when the PDF document is opened on the first page.

```

55     var iniocgtoolsOCGs = this.getOCGs();
56     var ocgtoolsOCGs = [];
57     for(var i=0; iniocgtoolsOCGs && i<iniocgtoolsOCGs.length;i++)
58       {
59         if(iniocgtoolsOCGs[i].name.substr(0,8) == "ocgtools")
60           {
61             ocgtoolsOCGs.push(iniocgtoolsOCGs[i]);
62             \if@ocgtools@insertvisible
63             iniocgtoolsOCGs[i].state=false;
64             \fi
65           }
66       }
67     iniocgtoolsOCGs[0].state=true;
68     iniocgtoolsOCGs[0].state=false;

```

```

69 \ifx\ocgtools@progressmsg\relax\relax\else
70 this.getField("ocgtoolsmsg").hidden=true;
71 \fi
72 }
73 \ifocg@hide@button
74 this.getField("OcgtoolsBtn.HideButton.main").hidden = true;
75 \fi
76 )
77 >> >>
78 }
79
80 \expandafter\global\expandafter\pdfpageattr\expandafter{\ocgtools@pdfpageattr}
81
82 \def\ocgtools@transparent{}
83 \if\ocgtools@transparent
84 \RequirePackage{transparent} \def\ocgtools@transparent{\transparent{0.4}}
85 \fi

```

Internal variables

```

86 \newif\ifocg@minitext@left
87 \newif\ifocg@minitext@bottom
88 \newcount\ocgtools@layercount
89 \newskip\ocgtools@left@skip
90 \newskip\ocgtools@bottom@skip
91 \newdimen\ocgtools@maxheight
92 \newdimen\ocgtools@maxwidth

```

We introduce lengths which can be used to finetune position of layers on the screen. These lengths are not necessary for plain pages (like article class with geometry package) and beamer.

```

93 \newdimen\layerHshift \layerHshift=0pt
94 \newdimen\layerVshift \layerVshift=0pt

```

We need small shift for web.sty.

```

95 \ifx\@Rightmargin\undefined\else
96 \layerVshift=3pt
97 \fi

```

We need another shift for pdfscreen.sty.

```

98 \ifx\PDFSCR@Warning\undefined\else
99 \layerVshift=\@Bottommargin
100 \fi
101

```

```

102 \newtoks\ocgtools@layer@toks
103 \ocgtools@layer@toks{}
104 \newbox\ocgtools@box@a
105 \newbox\ocgtools@box@b
106 \newif\ifocg@page@contains@layer

```

We modify shipout routine and insert content of layers on the top of PDF pages.

```

107 \def\ocgtools@one{1}
108 \let\ocgtools@shipoutstart@hook\relax
109 \let\ocgtools@shipoutend@hook\relax
110 \AtBeginShipout{%
111 \setbox\AtBeginShipoutBox=\hbox{%

```

Hook which can be used to place background.

```

112 \ocgtools@shipoutstart@hook

```

We insert the page first.

```

113 \hbox to 0 pt{\box\AtBeginShipoutBox\hss}\kern -1in\kern \layerHshift

```

We insert the layers.

```
114 \ocg@page@contains@layerfalse
115 \lower \layerVshift \hbox{\the\ocgtools@layer@toks}%
```

If at least one layer has been inserted, we insert button which can be used to hide layers.

```
116 \ifocg@page@contains@layer
117 \vbox to 0 pt{\kern -\paperheight \kern\layerVshift\hbox to 0 pt{\ocgtools@HideBtn\hss}\vss}%
118 \fi
```

We insert progress field on the first page.

```
119 \xdef\ocgtools@currpage{\thepage}%
120 \ifx\ocgtools@currpage\ocgtools@one
121 \ocgtools@progressmsg
122 \fi
```

We finish the box.

```
123 \hss \ocgtools@shipoutend@hook%
124 \kern 1in \kern -\layerHshift%
125 }%
126 }
```

We create buttons which are used to hide all layers (if we create buttons, we keep possibility to make them hidden, in contrast to links).

```
127 \def\ocgtools@HideBtn{\pushButton[\W{0}\BG{ }\S{S}\BC{ }\H{N}\F{\FHHidden}\A{\JS{
128     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
129     {
130         ocgtoolsOCGs[i].state = false;
131     }
132     \ifocg@hide@button this.getField("OcgttoolsBtn.HideButton").hidden = true; \fi
133 }]{OcgttoolsBtn.HideButton.main}{\paperwidth}{\paperheight}}
```

We create buttons which are attached to the top right corner of each minilayer and can be used to hide the corresponding minilayer.

```
134 \newcommand\ocgtools@HideMiniLayer[1]
135 {\setbox\ocgtools@box@a=\hbox{{\color{red}\ocgclosechar}}}%
136 \hbox to \wd\ocgtools@box@a%
137 {\hbox{{\color{red}\ocgclosechar}}\hss%
138 \hbox{\pushButton[\W{0}\S{S}\BG{ }\BC{ }\H{N}%
139 \F{\FHHidden}\A{\JS{
140     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
141     {
142         if(ocgtoolsOCGs[i].name == "ocgtools#1")
143         ocgtoolsOCGs[i].state = false;
144     }
145     this.getField("OcgttoolsBtn.HideButton.mini.#1").hidden = true;
146 }]]%
147 {OcgttoolsBtn.HideButton.mini.#1}{\wd\ocgtools@box@a}{\ht\ocgtools@box@a}}}
```

We create buttons which are attached to the bottom right corner of the screen and can be used to hide all layers.

```
148 \newcommand\ocgtools@HideLayers[1]{%
149 \setbox\ocgtools@box@a=\hbox{{\color{red}\ocgclosechar}}}%
150 \hbox to \wd\ocgtools@box@a%
151 {\hbox{{\color{red}\ocgclosechar}}\hss%
152 \hbox{\pushButton[\W{0}\S{S}\BG{ }\BC{ }\H{N}%
153 \F{\FHHidden}\A{\JS{
154     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
155     {
156         ocgtoolsOCGs[i].state = false;
157     }
158 }]]%
159 {OcgttoolsBtn.HideButton.all}{\wd\ocgtools@box@a}{\ht\ocgtools@box@a}}}
```

```

158         \ifocg@hide@button
159         this.getField("OcgttoolsBtn.HideButton.main").hidden = true;
160         \fi
161     }]}%
162     {OcgttoolsBtn.HideButton.corner.#1}{\wd\ocgttools@box@a}{\ht\ocgttools@box@a}}}}

```

To hide/unhide layers and minilayers we use modified code from <http://www.texample.net/weblog/2008/nov/02/creating-pdf-layers/>

```

163 \newcommand{\ocgttools@ToggleLayer}[2]{%
164   % #1: layer name,
165   % #2: link text
166   \leavevmode%
167   \pdfstartlink user {
168     /Subtype /Link
169     /Border [0 0 0]%
170     /H /N
171     /A <<
172     /S/JavaScript
173     /JS (
174       for(var i=0; ocgttoolsOCGs && i<ocgttoolsOCGs.length;i++)
175       {
176         if(ocgttoolsOCGs[i].name == "ocgttools#1")
177         ocgttoolsOCGs[i].state = !ocgttoolsOCGs[i].state;
178         else
179         ocgttoolsOCGs[i].state = false;
180       }
181       \ifocg@hide@button this.getField("OcgttoolsBtn.HideButton.main").hidden = false;\fi
182       this.getField("OcgttoolsBtn.HideButton.corner").hidden = false;
183       this.getField("OcgttoolsBtn.HideButton.corner.#1").setFocus();
184     )
185     >>
186   }#2%
187   \if@ocgttools@mouseover
188   \hbox to 0 pt{\hss\pushButton[\W{0}\S{S}\BG{}\BC{}\H{N}]%
189   \A{\JS{
190     var OpenedByMouseEnter#1=false;
191     \ifocg@hide@button this.getField("OcgttoolsBtn.HideButton.main").hidden = false;\fi
192     this.getField("OcgttoolsBtn.HideButton.corner").hidden = false;
193     this.getField("OcgttoolsBtn.HideButton.corner.#1").setFocus();
194   }}
195   \AA{\AAMouseEnter{\JS{      for(var i=0; ocgttoolsOCGs && i<ocgttoolsOCGs.length;i++)
196     {
197       var OpenedByMouseEnter#1=true;
198       if(ocgttoolsOCGs[i].name == "ocgttools#1")
199       ocgttoolsOCGs[i].state = true;
200       else
201       ocgttoolsOCGs[i].state = false;
202     }
203   }}
204   \AAMouseExit{\JS{
205     if (OpenedByMouseEnter#1)
206     {for(var i=0; ocgttoolsOCGs && i<ocgttoolsOCGs.length;i++)
207     {
208       if(ocgttoolsOCGs[i].name == "ocgttools#1")
209       ocgttoolsOCGs[i].state = false;
210     }
211   }}}
212   {OcgttoolsBtn.MaxiButton.#1}{8pt}{8pt}}}%
213   \fi

```



```

214 \pdfendlink%
215 }
216
217 \newcommand{\ocgtools@ToggleMiniLayer}[2]{%
218 % #1: layer name,
219 % #2: link text
220 \leavevmode%
221 \pdfstartlink user {
222 /Subtype /Link
223 /Border [0 0 0]%
224 /H /N
225 /A <<
226 /S/JavaScript
227 /JS (
228 for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
229 {
230 if(ocgtoolsOCGs[i].name == "ocgtools#1")
231 ocgtoolsOCGs[i].state = !ocgtoolsOCGs[i].state;
232 }
233 this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden =
234 !this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden;
235 if (!this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden)
236 {this.getField("OcgtoolsBtn.HideButton.mini.#1").setFocus();}
237 )
238 >>
239 }#2%
240 \if@ocgtools@minimouseover
241 \hbox to 0 pt{\hss\pushButton[\W{0}\S{S}\BG{}\BC{}\H{N}%
242 \A{\JS{
243 var OpenedByMouseEnter#1=false;
244 this.getField("OcgtoolsBtn.HideButton.mini.#1").setFocus();
245 }}
246 \AA{\AAMouseEnter{\JS{for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
247 {
248 var OpenedByMouseEnter#1=true;
249 if(ocgtoolsOCGs[i].name == "ocgtools#1")
250 ocgtoolsOCGs[i].state = true;
251 }
252 this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden = false;
253 }}
254 \AAMouseExit{\JS{
255 if (OpenedByMouseEnter#1)
256 {
257 for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
258 {
259 if(ocgtoolsOCGs[i].name == "ocgtools#1")
260 ocgtoolsOCGs[i].state = false;
261 }
262 this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden = true;
263 }
264 }}}]
265 {OcgtoolsBtn.MiniButton.#1}{8pt}{8pt}}}%
266 \fi
267 \pdfendlink%
268 }
269

```

Macros dealing with \savepos from Vit Zyka's paper in CSTUG Bulletin (ISSN 1211-6661, No. 2, 2007)

```

270 \newwrite\posHandle
271 \def\posFile{\jobname.pos}
272 \def\posOpen{\openout\posHandle=\posFile}
273 \def\posClose{\closeout\posHandle}
274 \ifocgtools@noocg
275 \else
276 \AtBeginDocument{\InputIfFileExists{\posFile}{-}{-}}%
277 \ocgtools@maxheight=0.9\paperheight
278 \ocgtools@maxwidth=0.9\paperwidth
279 \expandafter\global\expandafter\def\csname ocgtools@textcontent@0\endcsname
280 {\color{red}\bfseries\ocgtools@msg}
281 \expandafter\ocgtools@layer@toks\expandafter{\the\ocgtools@layer@toks}
282 \ocg@place@text{0}{1}{5cm}}
283 \posOpen}
284 \AtEndDocument{\posClose}
285 \fi
286

```

The following commands create links to hide/unhide layers and minilayers and write information on the layer number, page and layer content into the pos file. The content of textlayer is stored in command \ocgtools@textcontent@<n> where <n> is the number of the layer. Similarly, the content of the minitextlayer is stored in \ocgtools@minitextcontent@<n>.

```

287 \newcommand\ocgpicture[2][{}]{%
288 \global\advance\ocgtools@layercount by 1%
289 \ocgtools@ToggleLayer{\the\ocgtools@layercount}%
290 {\leavevmode\includegraphics[#1]{#2}}%
291 \expandafter\write\expandafter\posHandle\expandafter{%
292 \expandafter\string\expandafter\ocgtools@placepicture\expandafter{%
293 \the\ocgtools@layercount}{\the\page}{#2}}%
294 }
295
296 \newcommand\ocgtext[3][Opt]{%
297 \global\advance\ocgtools@layercount by 1%
298 \ocgtools@ToggleLayer{\the\ocgtools@layercount}%
299 {\leavevmode{\ocgtextstart #2\ocgtextend}}%
300 \expandafter\write\expandafter\posHandle\expandafter{%
301 \expandafter\string\expandafter\ocgtools@placetext\expandafter{%
302 \the\ocgtools@layercount}{\the\page}{#1}}%
303 \expandafter\global\expandafter\def
304 \csname ocgtools@textcontent@\the\ocgtools@layercount\endcsname{#3}%
305 }
306
307 \newcommand\ocgminitextlt[3][Opt]{%
308 \ocg@minitext@lefttrue\doocg@minitext{#1}{#2}{#3}{lt}}%
309 \newcommand\ocgminitextrt[3][Opt]{%
310 \ocg@minitext@leftfalse\doocg@minitext{#1}{#2}{#3}{rt}}%
311 \newcommand\ocgminitextlb[3][Opt]{%
312 \ocg@minitext@lefttrue\doocg@minitext{#1}{#2}{#3}{lb}}%
313 \newcommand\ocgminitextrb[3][Opt]{%
314 \ocg@minitext@leftfalse\doocg@minitext{#1}{#2}{#3}{rb}}%
315 \let\ocgminitext\ocgminitextrt
316
317 \long\def\doocg@minitext#1#2#3#4{%
318 \global\advance\ocgtools@layercount by 1\relax%
319 \leavevmode
320 \ifocg@minitext@left
321 \vbox to 0 pt{\kern 0.5\baselineskip\hbox to 0 pt{\hss\pdfsavepos}\vss}%
322 \fi
323 \ocgtools@ToggleMiniLayer{\the\ocgtools@layercount}%

```

```

324 {{\ocgtextstart #2\ocgtextend}}}%
325 \ifocg@minitext@left\else\pdfsavepos\fi
326 \expandafter\write\expandafter\posHandle\expandafter{%
327 \expandafter\string\expandafter\ocgtoolsplaceminitext\expandafter
328 {\the\ocgtools@layercount}{\thepage}{\the\pdfastxpos}%
329 {\the\pdfastypos}{\the\pdfastypos}{\the\pdfastypos}%
330 \expandafter\global\expandafter\def
331 \csname ocgtools@minitextcontent@{\the\ocgtools@layercount\endcsname}{#3}%
332 }%
333

```

These commands appear in `pos` file and we read these commands at the begin of the document. Pictures are stored in token register, texts for layers and minilayers are stored in commands defined by `\csname`.

```

334 \newcommand\ocgtoolsplacepicture[3]{%
335 \expandafter\global\expandafter\ocgtools@layer@toks\expandafter
336 {\the\ocgtools@layer@toks \ocg@place@picture{#1}{#2}{#3}}
337
338 \newcommand\ocgtoolsplacetext[3]{%
339 \expandafter\global\expandafter\ocgtools@layer@toks\expandafter
340 {\the\ocgtools@layer@toks \ocg@place@text{#1}{#2}{#3}}
341
342 \newcommand\ocgtoolsplaceminitext[7]{%
343 \expandafter\global\expandafter\ocgtools@layer@toks\expandafter
344 {\the\ocgtools@layer@toks \ocg@place@minitext{#1}{#2}{#3}{#4}{#5}{#6}{#7}}
345

```

These commands are called in output routine for each layer on each page. They put the layer on the page, if the layer should be here (i.e., if the second parameter equals `\thepage`).

```

346 \def\ocg@place@picture#1#2#3{\def\tempnuma{#2}\edef\tempnumb{\thepage}%
347 \ifx\tempnumb\tempnuma
348   \global\ocg@page@contains@layertrue
349   \vbox to 0 pt{\vss\hbox to 0 pt%
350     {\hbox{\begin{ocg}{\ocgtools#1}{\ocgtools#1}{\ocgtools@initialvisibility}%
351       \ocgtools@drawpicture{#3}{#1}\end{ocg}}\hss}}%
352 \fi}
353
354 \def\ocg@place@text#1#2#3{\def\tempnuma{#2}\edef\tempnumb{\thepage}%
355 \ifx\tempnumb\tempnuma
356   \global\ocg@page@contains@layertrue
357   \vbox to 0 pt{\vss\hbox to 0 pt{\hbox{%
358     \begin{ocg}{\ocgtools#1}{\ocgtools#1}{\ocgtools@initialvisibility}%
359     \ocgtools@drawtext{#3}{\csname ocgtools@textcontent@#1\endcsname}{#1}%
360     \end{ocg}}\hss}}%
361 \fi}
362
363 \long\def\ocg@place@minitext#1#2#3#4#5#6#7{%
364 \def\tempnuma{#2}\edef\tempnumb{\thepage}%
365 \def\ocg@placement{#7}%
366 \def\ocg@rb{rb}\def\ocg@lb{lb}\def\ocg@rt{rt}\def\ocg@lt{lt}%
367 \ocg@minitext@leftfalse\ocg@minitext@bottomfalse
368 \ifx\ocg@placement\ocg@lb
369   \ocg@minitext@lefttrue\ocg@minitext@bottomtrue
370 \fi
371 \ifx\ocg@placement\ocg@rb
372   \ocg@minitext@leftfalse\ocg@minitext@bottomtrue\fi
373 \ifx\ocg@placement\ocg@lt
374   \ocg@minitext@lefttrue\ocg@minitext@bottomfalse
375 \fi

```

```

376 \ifx\tempnumb\tempnuma
377 \ocgtools@left@skip=#3sp minus #3sp%
378 \ocgtools@bottom@skip=#4sp%
379 \setbox\ocgtools@box@a=\hbox{\begin{ocg}{\ocgtools#1}{\ocgtools#1}{\ocgtools@initialvisibility}%
380 \ocgtools@drawminitext{#6}{\csname ocgtools@minitextcontent@#1\endcsname}{#1}
381 \end{ocg}}}%
382 \ifocg@minitext@left
383 \ifdim \ocgtools@left@skip>\wd\ocgtools@box@a
384 \advance \ocgtools@left@skip by -\wd\ocgtools@box@a plus 0 pt minus -\wd\ocgtools@box@a
385 \else
386 \ocgtools@left@skip=0pt\relax
387 \fi
388 \fi
389 \ifocg@minitext@bottom
390 \advance \ocgtools@bottom@skip by -\ht\ocgtools@box@a
391 \fi
392 \hbox to 0 pt{\hbox to \paperwidth{\hskip \ocgtools@left@skip
393 \vbox to 0 pt{\vss
394 \vbox to \paperheight{\vskip 0 pt plus 1 fill
395 \box\ocgtools@box@a\vskip \ocgtools@bottom@skip}
396 }%
397 \hskip 0 pt plus 1 fill}\hss}%
398 \fi
399 }
400

```

This code actually gives a graphical representation of the layers.

```

401 \def\ocgtools@drawpicture#1#2{\vbox to \paperheight{\vbox to 0 pt{{%
402 \ocgtools@transparent\color{ocgbg}%
403 \hrule width \paperwidth height \paperheight}\vss}\vss
404 \hbox to \paperwidth{\hss
405 \setbox\ocgtools@box@a=\hbox{\includegraphics[height=\ocgtools@maxheight]{#1}}%
406 \ifdim\wd\ocgtools@box@a>\ocgtools@maxwidth
407 \includegraphics[width=\ocgtools@maxwidth]{#1}\else \box\ocgtools@box@a
408 \fi
409 \hss}%
410 \vss
411 \hbox to \paperwidth {\hss\ocgtools@HideLayers{#2}}}}
412
413 \newdimen\ocg@textdimen
414
415 \long\def\ocgtools@drawtext#1#2#3{\vbox to \paperheight{%
416 \vbox to 0 pt{{\ocgtools@transparent\color{ocgbg}%
417 \hrule width \paperwidth height \paperheight}\vss}\vss\hbox to \paperwidth
418 {\hss
419 \ocg@textdimen=#1
420 \ifdim\ocg@textdimen=0pt
421 \setbox\ocgtools@box@a=\hbox{\colorbox{ocgpaper}{#2}}}%
422 \else
423 \setbox\ocgtools@box@a=\hbox{\colorbox{ocgpaper}%
424 {\vbox{\hsize=#1\relax\rightskip 0 pt plus 1 fil\relax #2}}}%
425 \fi
426 \setbox\ocgtools@box@b\hbox{\resizebox{!}{\ocgtools@maxheight}{\copy\ocgtools@box@a}}%
427 \ifdim\wd\ocgtools@box@b>\ocgtools@maxwidth
428 \resizebox{\ocgtools@maxwidth}{!}{\box\ocgtools@box@a}
429 \else
430 \box\ocgtools@box@b
431 \fi
432 \hss}%

```

```

433 \vss
434 \hbox to \paperwidth {\hss\ocgtools@HideLayers{#3}}}}
435
436 \def\ocg@empty{
437
438 \newcommand\ocgtools@drawminitext[3]{\ocg@textdimen=#1
439 \ifdim\ocg@textdimen=0pt
440   \setbox\ocgtools@box@a=\hbox{\colorbox{ocgpaper}{#2\ }}%
441 \else \setbox\ocgtools@box@a=\hbox{\colorbox{ocgpaper}%
442   {\vbox{\hsize=#1\relax\rightskip 0 pt plus 1 fil\relax #2}}}%
443 \fi
444 \def\temp{#3}%
445 \ifx\temp\ocg@empty
446 \box\ocgtools@box@a
447 \else
448 \hbox{\copy\ocgtools@box@a\raise\ht\ocgtools@box@a
449 \vbox to 0 pt{\hbox to 0 pt{\hss
450 \ocgtools@HideMiniLayer{#3}}\vss}}}%
451 \fi
452 }
453
454 \ifocgtools@noocg
455 \renewcommand\ocgpicture[2][\leavevmode\includegraphics{#1}{#2}]
456 \renewcommand\ocgtext[3][0pt]{\leavevmode #2}
457 \long\def\doocg@minitext#1#2#3#4{\leavevmode #2}%
458 \renewcommand\ocgtoolsplacepicture[3]{}
459 \renewcommand\ocgtoolsplacetext[3]{}
460 \renewcommand\ocgtoolsplaceminitext[7]{}
461 \fi
462
463 \end{package}

```