

The `ocgtools` package^{*†}

Robert Mařík
marik@mendelu.cz

December 16, 2010

1 Introduction

The package `ocgtools` is designed to insert OCG (Optional Content Group) into PDF presentations. From the user's point of view, the package allows to insert in an comfortable way any \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 \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 page¹. `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 post-processing (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.

Important
comment!

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 (for known problems not related to portability see the end of this manual):

- `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.

^{*}This document corresponds to `ocgtools` v0.8a, dated 2010/11/19.

[†]Supported by grant 131/2010 of Higher Education Development Fund (FRVŠ)

¹remember to load `fancytooltips` as the first package

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

3.1 Compilation

All the packages we use are on CTAN or in TeXlive2009. The file `ocg.sty` is a part of `asymptote` package². AcroTeX is quite old on CTAN but the version on CTAN works well. New version of AcroTeX is at 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.

3.2 Package options

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

`transparent` Layers produced by `\ocgtext` and `\ocgpicture` in fact do not span over the whole page, but the material of layer is scaled to some reasonable size (we keep aspect ratio and the `\ocgtools@maxheight` width and height are not bigger than `\ocgtools@maxheight` and `\ocgtools@maxwidth`). If `\ocgtools@maxwidth` 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 like the following three lines to override:

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

Default values for `\ocgtools@maxheight` and `\ocgtools@maxwidth` are 0.9 of `\paperheight` and `\paperwidth`. You can redefine them after `\begin{document}` (remember to use `\makeatletter ... \makeatother` pair).

`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

²Do 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>

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 T_EX. 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.

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 and can be closed by hovering and exiting minibutton, clicking the active area outside the minibutton or clicking the red cross. The field with red cross gets focus after opening the layer and hence pressing Enter key also hides the layer or minilayer.

minimouseover As mouseover, but restricted to minilayers.

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.

inactive This option makes the package inactive.

noocg The same as **inactive**.

active This option forces the package active even if **inactive** option is loaded.

3.3 Text layer which extends to papersize

\ocgtext The macro `\ocgtext[width=<width>, bg=<color1>, fg=<color2>]{<text1>}{<text2>}` is used to create layer which contains any L^AT_EX material. *<text1>* 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 *<text2>* (which could be text divided into more than one paragraph, figure created by `mfpic` or any L^AT_EX material). *<text2>* is either placed into `\hbox` (if *<width>* is 0pt, which is default) or wrapped by `\vbox` with `\hsize` equal to *<width>* (in the opposite case). The color *<color1>* is used to set the background for this box and the color *<color2>* is used as text color. The default value for *<color1>* is stored in macro `\defaultocgpapercolor` and the default value for this macro is `ocgpaper`. The color `ocgpaper` is yellow by default, more precisely, it is declared with `\definecolor{ocgpaper}{rgb}{1,1,0.2}`. The default value for *<color2>* is stored in macro `\defaultocgfontcolor` and the default value for this macro is `ocgfontcolor`. The color `ocgfontcolor` is declared as black by default. You can use also key words `background` instead of `bg` and `color` instead of `fg`. The keyword `width` can be omitted, provided *<width>* comes as first argument. Thus `\ocgtext[1cm,background=black,color=white]{text}` puts the white text on black background in the box of width 1cm. For more details about color management see paragraph 3.7.

3.4 Layer with image which extends to papersize

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

3.5 Layer with text (or whatever) in its natural size

`\ocgminitext` The macro `\ocgminitext[width=<width>, bg=<color1>, fg=<color2>]{<text1>}{<text2>}` is used to create layer containing `<text2>` which is placed near the right top corner of `<text1>`. `<text2>` is placed into `\hbox` or `\vbox` with given `\hsize` according to the value of `<width>` and other optional parameters, 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.

Starting from version 0.9 you can use keywords `jsopen=<openaction>` and `jsclose=<closeaction>` to define JavaScript action which is performed when the layer is opened and closed, respectively.

3.6 Fine-tuning, customizing

`\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.

`ocgbg` The color `ocgbg` is used to set the color which are used to hide page when using `\ocgtext` and `\ocgpicture` commands. The default setting is `\definecolor{ocgbg}{rgb}{0,0,0}`. You may want to redefine this color, but you have to do this before `\begin{document}`.

`\layerHshift` The dimensions `\layerHshift` and `\layerVshift` are used to place layers exactly on the top of the page. Should be set automatically in the second pass, when reading aux file. If not (the big layers do not cover the PDF page and minitext layers are shifted), you may adjust them as required after `\begin{document}`. In this case, *report the problem* to the author of the package, please.

`\layerVshift` Commands `\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.

`\dots@envelope` The arguments of commands `\ocgpicture`, `\ocgtext` and family of `\ocgminitext` etc. are wrapped by macros `\ocgtools@pict@envelope`, `\ocgtools@text@envelope` and `\ocgtools@minitext@envelope`, respectively. You can redefine these macros to suit more to your need. All macros are `\relax` by default.

`\ocgclosechar` The `\ocgclosechar` command is used to print mark which is used to close layers. This mark is a red cross. On the minitextlayers with red background we switch this color to black (see the first few lines of the code to see, how we get this behavior). Note that we used the macro `\if@ocgtools@insidemini@layer` which is true on layers with minitext and false otherwise.

`\ocgtools@msg` The `\ocgtools@msg` command contains string for users of viewers which do not support layers (see the first few lines of the code for default settings).

3.7 Summary on customizing colors in text layers

Note that all colors related to the layers are set (and hence must be declared before) when reading auxiliary files at `\begin{document}`. Declaring of redefining colors after `\begin{document}` has no influence. To summarize, if you want to change colors, use the following.

- To change the setting in the whole paper, redeclare colors `ocgpaper` and `ocgfontcolor` before `\begin{document}`. Use the command `\definecolor` for this purpose.
- To change the setting from now to the end of document (or group) redefine macros `\defaultocgpapercolor` and `\defaultocgfontcolor`. Use for example `\renewcommand{\defaultocgpapercolor}{green}`.
- To change the setting for one single layer use `bg` and `fg` keywords in optional parameter of `\ocgtext` and `\ocgminitext` macros.

4 Possible future development and known problems

The source code is in Mercurial repository at <http://bitbucket.org/robert.marik/ocgtools/>. You can also report problems and issues in the forum at this site.

4.1 Known problems

1. The package does not work properly if the **PDF has one page** only. This is probably minor problem, since most presentations are longer and the problem will be traced later.
2. There could be a **conflict with another package which deals with `\pdfpageattr`**. This problem is resolved for `fancytooltips` package. To make both `fancytooltips` and `ocgtools` work in one document, load *fancytooltips as the first one* and `ocgtools` after. Remember the only one `\pdfpageattr` is allowed in PDF specification for each single page.
3. In some rare cases the layers may be not inserted properly when the **L^AT_EX file is compiled on Windows** – see the option `insertvisible` which solves this problems in most cases. Since the author has limited access to T_EX installations on Windows and hence reporting of problems with a minimal example is highly appreciated.
4. 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.

4.2 Ideas for future development

These ideas may appear in new versions of the package (and patches which include the solutions to this or other problems are welcomed).

1. 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> and the demo `examples/fancy-preview` in `fancytooltips` distribution. See the `ocgtools-preview.*` files for some initial attempts in this direction based on `ocgtools`.
2. Introduce draft mode, which prints all layers at the end of document with links there and back?
3. 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 this 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{ocgfontcolor}{rgb}{0,0,0}
12 \def\defaultocgpapercolor{ocgpaper}
13 \def\defaultocgfontcolor{ocgfontcolor}
14 \definecolor{ocgbg}{rgb}{0,0,0}
15 \RequirePackage{graphicx}
16 \RequirePackage{pifont}
17 \RequirePackage{ocg}
18 \RequirePackage{xkeyval}
19
20 \newif\if@ocgtools@insidemini@layer
21 \def\ocgclosechar{{\color{red}\def\temp{red}%
22   \ifx\temp\ocgt@ls@bg\if@ocgtools@insidemini@layer\color{black}\fi\fi
23   \ding{56}}}}
24
```

Make packages and options known.

```
25 \RequirePackage{atbegshi}
26 \RequirePackage[pdftex]{eforms}
27 \newif\if@ocgtools@transparent\@ocgtools@transparentfalse
28 \DeclareOption{transparent}{\@ocgtools@transparenttrue}
29 \newif\if@ocgtools@insertvisible\@ocgtools@insertvisiblefalse
30 \def\ocgtools@initialvisibility{0}
31 \DeclareOption{insertvisible}{\@ocgtools@insertvisibletrue}
32   \def\ocgtools@initialvisibility{1}}
33 \newif\ifocg@hide@button\ocg@hide@buttontrue
34 \DeclareOption{nobutton}{\ocg@hide@buttonfalse}
35 \def\ocgtools@progressmsg{\lower \layerVshift\hbox to 0 pt{ %space
36   \textField[\V{OCGtools: processing OCG's ...}\BG{1 1 1}
37   \textColor{1 0 0} \textSize{10}}}%
38 {ocgtoolsmsg}{6cm}{20pt}\hss}}%
39 \DeclareOption{noprogessmsg}{\let\ocgtools@progressmsg\relax}
40 \newif\if@ocgtools@minimouseover \@ocgtools@minimouseoverfalse
41 \DeclareOption{minimouseover}{\@ocgtools@minimouseovertrue}
42 \newif\if@ocgtools@mouseover \@ocgtools@mouseoverfalse
43 \DeclareOption{mouseover}{\@ocgtools@mouseovertrue \@ocgtools@minimouseovertrue}
44 \newif\if@ocgtools@pageclose \@ocgtools@pageclosetrue
45 \DeclareOption{nopageclose}{\@ocgtools@pageclosefalse}
46 \newif\if@ocgtools@inactive\@ocgtools@inactivefalse
47 \DeclareOption{inactive}{\@ocgtools@inactivetrue}
48 \DeclareOption{noocg}{\@ocgtools@inactivetrue}
49 \newif\if@ocgtools@active\@ocgtools@activefalse
50 \DeclareOption{active}{\@ocgtools@activetrue}
51
52 \ProcessOptions
53
54 \if@ocgtools@active\@ocgtools@inactivefalse\fi
```

```

55 \newdimen\layerHshift \layerHshift=-1in
56 \newdimen\layerVshift \layerVshift=0pt
57
58 \if@ocgtools@inactive
59 \newcommand\ocgpicture[2] [] {\leavevmode\includegraphics[#1]{#2}}
60 \newcommand\ocgtext[3] [Opt] {\leavevmode #2}
61 \newcommand\ocgtoolsplacepicture[3] {}
62 \newcommand\ocgtoolsplacetext[3] {}
63 \newcommand\ocgtoolsplaceminitext[7] {}
64 \newcommand\ocgminitext[3] [Opt] {\leavevmode #2}%
65 \let\ocgminitextlt\ocgminitext
66 \let\ocgminitextrt\ocgminitext
67 \let\ocgminitextlb\ocgminitext
68 \let\ocgminitextrb\ocgminitext
69 \PackageWarning{ocgtools}{Ocgttools inactive}%
70 \expandafter\endinput\fi
71

```

We use the hack for Adobe Acrobat suggested by DPS and Jorg at <http://www.acrotex.net/forum/showthread.php?tid=78>

```

72 \def\ocgtools@JS#1{\JS{DirtyBeforeOCGtools=this.dirty; #1
73 this.dirty=DirtyBeforeOCGtools;}}

```

We insert JavaScripts which are evaluated when the file is opened at the first page. There is a clash with fancytooltips package. To make things work, load fancytooltips before ocgtools and add the \pdfpageattr from fancytooltips to \pdfpageattr inserted by ocgtools.

```

74 \@ifpackageloaded{fancytooltips}{%
75 \let\TooltipPageopencloseJS\relax
76 \ifx\fancytooltips@pdfpageattrJS\undefined
77 \def\@ocgtools@fancytooltips{var DirtyBeforeCloseTooltip=this.dirty;
78 CloseTooltips(); this.dirty=DirtyBeforeCloseTooltip;}
79 \else
80 \let\@ocgtools@fancytooltips\fancytooltips@pdfpageattrJS
81 \fi
82 }\def\@ocgtools@fancytooltips{}}
83
84 \edef\ocgtools@pdfpageattr{
85 /AA << /O << /S /JavaScript /JS
86 (

```

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.

```

87 if (typeof(ocgtoolsOCGs)!="undefined"){
88 var temp = ocgtoolsOCGs.length;
89 \if@ocgtools@pageclose
90 for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
91 {
92 ocgtoolsOCGs[i].state = false;
93 }
94 \fi
95 }

```

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

```

96 else {

```

No dotted rectangle for buttons which have focus.

```

97 app.focusRect = false;
98 var DirtyBeforeOCGtools=this.dirty;

```

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

```

99   var iniocgtoolsOCGs = this.getOCGs();
100  var ocgtoolsOCGs = [];
101  for(var i=0; iniocgtoolsOCGs && i<iniocgtoolsOCGs.length;i++)
102  {
103    if(iniocgtoolsOCGs[i].name.substr(0,8) == "ocgtools")
104    {
105      ocgtoolsOCGs.push(iniocgtoolsOCGs[i]);
106      \if@ocgtools@insertvisible
107      iniocgtoolsOCGs[i].state=false;
108      \fi
109    }
110  }
111  iniocgtoolsOCGs[0].state=true;
112  iniocgtoolsOCGs[0].state=false;
113  \ifx\ocgtools@progressmsg\relax\relax\else
114  this.getField("ocgtoolsmsg").hidden=true;
115  this.dirty=false;
116  \fi
117  }
118  \ifocg@hide@button
119  this.getField("OcgttoolsBtn.HideButton.main").hidden = true;
120  this.dirty=false;
121  \fi
122  \@ocgtools@fancytooltips
123  )
124  >> >>
125 }
126
127 \expandafter\global\expandafter\pdfpageattr\expandafter{\ocgtools@pdfpageattr}
128
129 \def\ocgtools@transparent{}
130 \if@ocgtools@transparent
131 \RequirePackage{transparent} \def\ocgtools@transparent{\transparent{0.4}}
132 \fi

```

Internal variables

```

133 \newif\ifocg@minitext@left
134 \newif\ifocg@minitext@bottom
135 \newcount\ocgtools@layercount
136 \newskip\ocgtools@left@skip
137 \newskip\ocgtools@bottom@skip
138 \newdimen\ocgtools@maxheight
139 \newdimen\ocgtools@maxwidth

```

We introduce lengths which can be used to fine-tune position of layers on the screen. These lengths should be set automatically.

```

140 \def\ocgtools@save@position{\pdfsavepos
141 \write\@auxout{\string\global\string\advance\string \layerVshift \the\pdflastypos sp\string\relax}%
142 \write\@auxout{\string\global\string\advance\string \layerHshift \the\pdflastxpos sp\string\relax}%
143 \global\let\ocgtools@save@position\relax
144 }
145
146 \newtoks\ocgtools@layer@toks
147 \ocgtools@layer@toks{\ocg@place@text{0}{1}{5cm}}
148 \newbox\ocgtools@box@a
149 \newbox\ocgtools@box@b
150 \newif\ifocg@page@contains@layer

```

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

```

151 \def\ocgtools@one{1}

```



```

152 \let\ocgtools@shipoutstart@hook\relax
153 \let\ocgtools@shipoutend@hook\relax
154 \AtBeginShipout{%
155 \expandafter\global\expandafter\pdfpageattr\expandafter{\ocgtools@pdfpageattr}
156 \setbox\AtBeginShipoutBox=\hbox{%

```

Hook which can be used to place background.

```
157 \ocgtools@shipoutstart@hook
```

This command writes the info about the position on the first PDF page and then is set to relax on next pages.

```
158 \ocgtools@save@position
```

We insert the page first.

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

We insert the layers.

```
160 \ocg@page@contains@layerfalse
```

```
161 \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.

```
162 \ifocg@page@contains@layer
```

```
163 \vbox to 0 pt{\kern -\paperheight \kern\layerVshift\hbox to 0 pt{\ocgtools@HideBtn\hss}\vss}%
```

```
164 \fi
```

We insert progress field on the first page.

```
165 \xdef\ocgtools@currpage{\thepage}%
```

```
166 \ifx\ocgtools@currpage\ocgtools@one
```

```
167 \ocgtools@progressmsg
```

```
168 \fi
```

We finish the box.

```
169 \hss \ocgtools@shipoutend@hook%
```

```
170 \kern 1in \kern -\layerHshift%
```

```
171 }%
```

```
172 }
```

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).

```
173 \ifx\ocgtools@AfterHideBtn\undefined\def\ocgtools@AfterHideBtn{}\fi
```

```
174 \def\ocgtools@HideBtn{\pushButton[\W{0}\BG{ }\S{S}\BC{ }\H{N}\F{\FHHidden}\A{\ocgtools@JS{
```

```
175     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
```

```
176     {
```

```
177     ocgtoolsOCGs[i].state = false;
```

```
178     }
```

```
179     \ifocg@hide@button this.getField("OcgttoolsBtn.HideButton").hidden = true; \fi
```

```
180     \ocgtools@AfterHideBtn
```

```
181     }]}{\ocgtoolsBtn.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.

```
182 \ifx\ocgtools@AfterHideMiniLayer\undefined\def\ocgtools@AfterHideMiniLayer{}\fi
```

```
183 \newcommand\ocgtools@HideMiniLayer[2] []
```

```
184 {\setbox\ocgtools@box@a=\hbox{{\ocgclosechar}}}%
```

```
185 \hbox to \wd\ocgtools@box@a%
```

```
186 {\hbox{{\ocgclosechar}}\hss%
```

```
187 \hbox{\pushButton[\W{0}\S{S}\BG{ }\BC{ }\H{N}%
```

```
188 \F{\FHHidden}\A{\ocgtools@JS{
```

```
189     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
```

```
190     {
```

```
191     if(ocgtoolsOCGs[i].name == "ocgtools#2")
```

```
192     ocgtoolsOCGs[i].state = false;
```

```

193   }
194   this.getField("OcgtoolsBtn.HideButton.mini.#2").hidden = true;
195   \ocgtools@AfterHideMiniLayer
196   #1
197   }]}%
198 {OcgtoolsBtn.HideButton.mini.#2}{\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.

```

199 \ifx\ocgtools@AfterHideLayers\undefined\def\ocgtools@AfterHideLayers{}\fi
200 \newcommand\ocgtools@HideLayers[1]{%
201 \setbox\ocgtools@box@a=\hbox{\ocgclosechar}}%
202 \hbox to \wd\ocgtools@box@a%
203 {\hbox{\ocgclosechar}\hss%
204 \hbox{\pushButton[\W{0}\S{S}\BG{} \BC{} \H{N}%
205 \F{\FHHidden}\A{\ocgtools@JS{
206     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
207     {
208         ocgtoolsOCGs[i].state = false;
209     }
210     \ifocg@hide@button
211     this.getField("OcgtoolsBtn.HideButton.main").hidden = true;
212     \fi
213     \ocgtools@AfterHideLayers
214     }]}%
215 {OcgtoolsBtn.HideButton.corner.#1}{\wd\ocgtools@box@a}{\ht\ocgtools@box@a}}

```

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

```

216 \newcommand{\ocgtools@ToggleLayer}[2]{%
217 % #1: layer name,
218 % #2: link text
219 \leavevmode%
220 \pdfstartlink user {
221 /Subtype /Link
222 /Border [0 0 0]%
223 /H /N
224 /A <<
225 /S/JavaScript
226 /JS (
227     DirtyBeforeOCGtools=this.dirty;
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         else
233             ocgtoolsOCGs[i].state = false;
234     }
235     \ifocg@hide@button this.getField("OcgtoolsBtn.HideButton.main").hidden = false;\fi
236     this.getField("OcgtoolsBtn.HideButton.corner").hidden = false;
237     this.getField("OcgtoolsBtn.HideButton.corner.#1").setFocus();
238     this.dirty=DirtyBeforeOCGtools;
239     )
240     >>
241 }#2%
242 \if@ocgtools@mouseover
243 \hbox to 0 pt{\hss\pushButton[\W{0}\S{S}\BG{} \BC{} \H{N}%
244 \A{\ocgtools@JS{
245     var OpenedByMouseEnter#1=false;

```

```

246     \ifocg@hide@button this.getField("OcgtoolsBtn.HideButton.main").hidden = false;\fi
247     this.getField("OcgtoolsBtn.HideButton.corner").hidden = false;
248     this.getField("OcgtoolsBtn.HideButton.corner.#1").setFocus();
249     }}
250 \AA{\AAMouseEnter{\ocgtools@JS{           for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
251     {
252         var OpenedByMouseEnter#1=true;
253         if(ocgtoolsOCGs[i].name == "ocgtools#1")
254             ocgtoolsOCGs[i].state = true;
255         else
256             ocgtoolsOCGs[i].state = false;
257     }
258     }}
259 \AAMouseExit{\ocgtools@JS{
260     if (OpenedByMouseEnter#1)
261         {for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
262         {
263             if(ocgtoolsOCGs[i].name == "ocgtools#1")
264                 ocgtoolsOCGs[i].state = false;
265         }
266     }}}}
267     {OcgtoolsBtn.MaxiButton.#1}{8pt}{8pt}}%
268 \fi
269 \pdfendlink%
270 }
271
272 \newcommand{\ocgtools@ToggleMiniLayer}[4]{%
273 % #1: layer name,
274 % #2: link text
275 % #3: js action if the layer is opened
276 % #4: js action if the layer is closed
277 \leavevmode%
278 \pdfstartlink user {
279 /Subtype /Link
280 /Border [0 0 0]%
281 /H /N
282 /A <<
283 /S/JavaScript
284 /JS (
285     DirtyBeforeOCGtools=this.dirty;
286     for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
287     {
288         if(ocgtoolsOCGs[i].name == "ocgtools#1")
289         {
290             if (ocgtoolsOCGs[i].state)
291             {
292                 #4
293                 ocgtoolsOCGs[i].state = false;
294             }
295             else
296             {
297                 #3
298                 ocgtoolsOCGs[i].state = true;
299             }
300         }
301     }
302     this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden =
303     !this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden;

```

```

304         if (!this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden)
305             {this.getField("OcgtoolsBtn.HideButton.mini.#1").setFocus();}
306         this.dirty=DirtyBeforeOCGtools;
307     )
308     >>
309 }#2%
310 \if@ocgtools@minimouseover
311 \hbox to 0 pt{\hss\pushButton[\W{0}\S{S}\BG{}BC{}H{N}%
312 \A{\ocgtools@JS{
313     var OpenedByMouseEnter#1=false;
314     this.getField("OcgtoolsBtn.HideButton.mini.#1").setFocus();
315 }}
316 \AA{\AAMouseEnter{\ocgtools@JS{for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
317     {
318         var OpenedByMouseEnter#1=true;
319         if(ocgtoolsOCGs[i].name == "ocgtools#1")
320             {#3 ocgtoolsOCGs[i].state = true;}
321     }
322     this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden = false;
323 }}
324 \AAMouseExit{\ocgtools@JS{
325     if (OpenedByMouseEnter#1)
326     {
327         for(var i=0; ocgtoolsOCGs && i<ocgtoolsOCGs.length;i++)
328         {
329             if(ocgtoolsOCGs[i].name == "ocgtools#1")
330                 {#4 ocgtoolsOCGs[i].state = false; }
331         }
332         this.getField("OcgtoolsBtn.HideButton.mini.#1").hidden = true;
333     }
334 }}}]
335 {OcgtoolsBtn.MiniButton.#1}{8pt}{8pt}}%
336 \fi
337 \pdfendlink%
338 }
339

```

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

```

340 \newwrite\posHandle
341 \def\posFile{\jobname.pos}
342 \def\posOpen{\openout\posHandle=\posFile}
343 \def\posClose{\closeout\posHandle}
344 \AtBeginDocument{\InputIfFileExists{\posFile}{}}%
345 \ocgtools@maxheight=0.9\paperheight
346 \ocgtools@maxwidth=0.9\paperwidth
347 \expandafter\global\expandafter\def\csname ocgtools@textcontent@0\endcsname
348 {{\color{red}\bfseries\ocgtools@msg}}%
349 \posOpen}
350 \AtEndDocument{\posClose}
351

```

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>.

```

352 \newcommand\ocgpicture[2][ ]{%
353 \global\advance\ocgtools@layercount by 1%
354 \ocgtools@ToggleLayer{\the\ocgtools@layercount}%

```

```

355 {\leavevmode\includegraphics[#1]{#2}}%
356 \expandafter\write\expandafter\posHandle\expandafter{%
357 \expandafter\string\expandafter\ocgtoolsplacepicture\expandafter{%
358 \the\ocgtools@layercount}\the\page}{#2}}%
359 }
360
361 \define@key{ocg@key}{width}[Opt]{\def\ocgt@@ls@width{#1}}
362 \define@key{ocg@key}{background}[ocgpaper]{\def\ocgt@@ls@bg{#1}}
363 \define@key{ocg@key}{bg}[ocgpaper]{\def\ocgt@@ls@bg{#1}}
364 \define@key{ocg@key}{color}[ocgfontcolor]{\def\ocgt@@ls@fg{#1}}
365 \define@key{ocg@key}{fg}[ocgfontcolor]{\def\ocgt@@ls@fg{#1}}
366 \define@key{ocg@key}{jsopen}[]{\def\ocgt@@ls@jsopen{#1}}
367 \define@key{ocg@key}{jsclose}[]{\def\ocgt@@ls@jsclose{#1}}
368
369 \newtoks\ocg@pdflastxpos\ocg@pdflastxpos{\the\pdflastxpos}
370 \newtoks\ocg@pdflastypos\ocg@pdflastypos{\the\pdflastypos}
371 \newtoks\ocg@pagetoks\ocg@pagetoks{\the\page}
372

```

The trick with `\setkeys*` allows to write `[2cm,fg=red]` instead of `[width=2cm,fg=red]`.

```

373 \def\ocgtools@setdefault{\def\ocgt@@ls@bg{\defaultocgpapercolor}}%
374 \def\ocgt@@ls@fg{\defaultocgfontcolor}%
375 \def\ocgt@@ls@width{Opt}\def\ocgt@@ls@jsopen{}%
376 \def\ocgt@@ls@jsclose{}%
377 }
378
379 \newcommand\ocgtext[3][Opt]{%
380 \ocgtools@setdefault
381 \setkeys*{ocg@key}{#1}%
382 \ifx\XKV@rm\empty\setkeys{ocg@key}{#1}\else\setkeys{ocg@key}{width=#1}\fi
383 \global\advance\ocgtools@layercount by 1%
384 \ocgtools@ToggleLayer{\the\ocgtools@layercount}%
385 {\leavevmode{\ocgtextstart #2\ocgtextend}}%
386 \edef\ocgtools@act{\write\posHandle{\noexpand\string\noexpand\ocgtoolsplacetext{\the\ocgtools@layercount}\the\page}}%
387 \ocgtools@act
388 \expandafter\global\expandafter\def
389 \csname ocgtools@textcontent@\the\ocgtools@layercount\endcsname{#3}%
390 }
391
392 \newcommand\ocgminitextlt[3][Opt]{%
393 \ocg@minitext@bottomfalse\ocg@minitext@lefttrue
394 \do@ocg@minitext{#1}{#2}{#3}{lt}}%
395 \newcommand\ocgminitextrt[3][Opt]{%
396 \ocg@minitext@bottomfalse\ocg@minitext@leftfalse
397 \do@ocg@minitext{#1}{#2}{#3}{rt}}%
398 \newcommand\ocgminitextlb[3][Opt]{%
399 \ocg@minitext@bottomtrue\ocg@minitext@lefttrue
400 \do@ocg@minitext{#1}{#2}{#3}{lb}}%
401 \newcommand\ocgminitextrb[3][Opt]{%
402 \ocg@minitext@bottomtrue\ocg@minitext@leftfalse
403 \do@ocg@minitext{#1}{#2}{#3}{rb}}%
404 \let\ocgminitext\ocgminitextrt
405
406 \long\def\do@ocg@minitext#1#2#3#4{%
407 \ocgtools@setdefault
408 \setkeys*{ocg@key}{#1}%
409 \ifx\XKV@rm\empty\setkeys{ocg@key}{#1}\else\setkeys{ocg@key}{width=#1}\fi
410 \global\advance\ocgtools@layercount by 1\relax%
411 \leavevmode

```

```

412 \ifocg@minitext@left
413 \ifocg@minitext@bottom
414 \vbox to 0 pt{\kern 0.7\baselineskip\pdfsavepos\vss}%
415 \else
416 \vbox to 0 pt{\vss\pdfsavepos\kern 0.7\baselineskip}%
417 \fi
418 \fi
419 \ocgtools@ToggleMiniLayer{\the\ocgtools@layercount}%
420 {\ocgtextstart #2\ocgtextend}{\ocgt@ls@jsopen}{\ocgt@ls@jsclose}%
421 \ifocg@minitext@left\else
422 \ifocg@minitext@bottom
423 \vbox to 0 pt{\kern 0.7\baselineskip\pdfsavepos\vss}%
424 \else
425 \vbox to 0 pt{\vss\pdfsavepos\kern 0.7\baselineskip}%
426 \fi
427 \fi
428 \edef\ocgtools@act{\write\posHandle{\noexpand\string\noexpand\ocgtoolsplaceminitext{\the\ocgtools@layercount}}
429 \ocgtools@act
430 \expandafter\global\expandafter\def
431 \csname ocgtools@minitextcontent@the\ocgtools@layercount\endcsname{#3}%
432 }%
433

```

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`.

```

434 \newcommand\ocgtoolsplacepicture[3]{%
435 \expandafter\global\expandafter\ocgtools@layer@toks\expandafter
436 {\the\ocgtools@layer@toks \ocg@place@picture{#1}{#2}{#3}}
437
438 \newcommand\ocgtoolsplacetext[3]{%
439 \expandafter\global\expandafter\ocgtools@layer@toks\expandafter
440 {\the\ocgtools@layer@toks\ocg@place@text{#1}{#2}{#3}}
441
442 \newcommand\ocgtoolsplaceminitext[7]{%
443 \expandafter\global\expandafter\ocgtools@layer@toks\expandafter
444 {\the\ocgtools@layer@toks \ocg@place@minitext{#1}{#2}{#3}{#4}{#5}{#6}{#7}}
445

```

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`).

```

446 \def\ocg@place@picture#1#2#3{\def\tempnuma{#2}\edef\tempnumb{\thepage}%
447 \ifx\tempnumb\tempnuma
448 \global\ocg@page@contains@layertrue
449 \vbox to 0 pt{\vss\hbox to 0pt%
450 {\hbox{\begin{ocg}{ocgtools#1}{ocgtools#1}{\ocgtools@initialvisibility}%
451 \ocgtools@drawpicture{#3}{#1}\end{ocg}}\hss}}%
452 \fi}
453
454 \def\ocg@place@text#1#2#3{\def\tempnuma{#2}\edef\tempnumb{\thepage}%
455 \ifx\tempnumb\tempnuma
456 \global\ocg@page@contains@layertrue
457 \vbox to 0 pt{\vss\hbox to 0 pt{\hbox{%
458 \begin{ocg}{ocgtools#1}{ocgtools#1}{\ocgtools@initialvisibility}%
459 \ocgtools@drawtext{#3}{\csname ocgtools@textcontent@#1\endcsname}{#1}%
460 \end{ocg}}\hss}}%
461 \fi}
462
463 \long\def\ocg@place@minitext#1#2#3#4#5#6#7{%

```

```

464 \def\tempnuma{#2}\edef\tempnumb{\thepage}%
465 \def\ocg@placement{#7}%
466 \def\ocg@rb{rb}\def\ocg@lb{lb}\def\ocg@rt{rt}\def\ocg@lt{lt}%
467 \ocg@minitext@leftfalse\ocg@minitext@bottomfalse
468 \ifx\ocg@placement\ocg@lb
469   \ocg@minitext@lefttrue\ocg@minitext@bottomtrue
470 \fi
471 \ifx\ocg@placement\ocg@rb
472   \ocg@minitext@leftfalse\ocg@minitext@bottomtrue\fi
473 \ifx\ocg@placement\ocg@lt
474   \ocg@minitext@lefttrue\ocg@minitext@bottomfalse
475 \fi
476 \ifx\tempnumb\tempnuma
477   \ocgtools@left@skip=#3sp minus #3sp%
478   \ocgtools@bottom@skip=#4sp%
479   \setbox\ocgtools@box@a=\hbox{\begin{ocg}{\ocgtools#1}{\ocgtools#1}{\ocgtools@initialvisibility}%
480   \ocgtools@drawminitext{#6}{\csname ocgtools@minitextcontent@#1\endcsname}{#1}
481   \end{ocg}}%
482   \ifocg@minitext@left
483     \ifdim \ocgtools@left@skip>\wd\ocgtools@box@a
484       \advance \ocgtools@left@skip by -\wd\ocgtools@box@a plus 0 pt minus -\wd\ocgtools@box@a
485     \else
486       \ocgtools@left@skip=0pt\relax
487     \fi
488   \fi
489   \ifocg@minitext@bottom
490     \advance \ocgtools@bottom@skip by -\ht\ocgtools@box@a
491   \fi
492   \hbox to 0 pt{\hbox to \paperwidth{\hskip \ocgtools@left@skip
493   \vbox to 0 pt{\vss
494   \vbox to \paperheight{\vskip 0 pt plus 1 fill
495   \box\ocgtools@box@a\vskip \ocgtools@bottom@skip}
496   }%
497   \hskip 0 pt plus 1 fill}\hss}%
498 \fi
499 }
500

```

This code actually gives a graphical representation of the layers.

```

501 \let\ocgtools@pict@envelope\relax
502 \def\ocgtools@drawpicture#1#2{\vbox to \paperheight{\vbox to 0 pt{
503 \ocgtools@transparent\color{ocgbg}%
504 \hrule width \paperwidth height \paperheight}\vss}\vss
505 \hbox to \paperwidth{\hss
506 \setbox\ocgtools@box@a=\hbox{\ocgtools@pict@envelope{
507 \includegraphics[height=\ocgtools@maxheight]{#1}}}%
508 \ifdim\wd\ocgtools@box@a>\ocgtools@maxwidth
509   \ocgtools@pict@envelope{\includegraphics[width=\ocgtools@maxwidth]{#1}}%
510   \else \box\ocgtools@box@a
511 \fi
512 \hss}%
513 \vss
514 \hbox to \paperwidth {\hss\ocgtools@HideLayers{#2}}}}
515
516 \newdimen\ocg@textdimen
517 \let\ocgtools@text@envelope\relax
518 \long\def\ocgtools@drawtext#1#2#3{%
519 \ocgtools@setdefault
520 \setkeys*{ocg@key}{#1}%

```

```

521 \ifx\XKV@rm\empty\setkeys{ocg@key}{#1}\else\setkeys{ocg@key}{width=#1}\fi
522 \vbox to \paperheight{%
523 \vbox to 0 pt{{\ocgtools@transparent\color{ocgbg}%
524 \hrule width \paperwidth height \paperheight}\vss}\vss\hbox to \paperwidth
525 {\hss
526 \ifdim\ocgt@@ls@width=0pt
527 \setbox\ocgtools@box@a=\hbox{\expandafter\colorbox\expandafter{\ocgt@@ls@bg}%
528 {\expandafter\color\expandafter{\ocgt@@ls@fg}\ocgtools@text@envelope{#2}}}%
529 \else
530 \setbox\ocgtools@box@a=\hbox{\expandafter\colorbox{\expandafter\ocgt@@ls@bg}%
531 {\vbox{\hsize=\ocgt@@ls@width\relax\linewidth=\ocgt@@ls@width\relax
532 \rightskip 0 pt plus 1 fil\relax
533 \expandafter\color\expandafter{\ocgt@@ls@fg}\ocgtools@text@envelope{#2}}}}}%
534 \fi
535 \setbox\ocgtools@box@b\hbox{\resizebox{!}{\ocgtools@maxheight}{\copy\ocgtools@box@a}}%
536 \ifdim\wd\ocgtools@box@b>\ocgtools@maxwidth
537 \resizebox{\ocgtools@maxwidth}{!}{\box\ocgtools@box@a}
538 \else
539 \box\ocgtools@box@b
540 \fi
541 \hss}%
542 \vss
543 \hbox to \paperwidth {\hss\ocgtools@HideLayers{#3}}%
544 }%
545 }
546
547 \def\ocg@empty{}
548
549 \let\ocgtools@minitext@envelope\relax
550 \newcommand\ocgtools@drawminitext[3]{%
551 \@ocgtools@insidemini@layertrue
552 \ocgtools@setdefault
553 \setkeys*{ocg@key}{#1}%
554 \ifx\XKV@rm\empty\setkeys{ocg@key}{#1}\else\setkeys{ocg@key}{width=#1}\fi
555 \ifdim\ocgt@@ls@width=0pt
556 \setbox\ocgtools@box@a=\hbox{\colorbox{\ocgt@@ls@bg}{\expandafter\color
557 \expandafter{\ocgt@@ls@fg}\ocgtools@minitext@envelope{#2}\ }}%
558 \else \setbox\ocgtools@box@a=\hbox{\colorbox{\ocgt@@ls@bg}%
559 {\vbox{\hsize=\ocgt@@ls@width\relax\linewidth=\ocgt@@ls@width\relax
560 \rightskip 0 pt plus 1 fil\relax
561 \expandafter\color\expandafter{\ocgt@@ls@fg}\ocgtools@minitext@envelope{#2}}}}}%
562 \fi
563 \def\temp{#3}%
564 \ifx\temp\ocg@empty
565 \box\ocgtools@box@a
566 \else
567 \hbox{\copy\ocgtools@box@a\raise\ht\ocgtools@box@a
568 \vbox to 0 pt{\hbox to 0 pt{\hss
569 \ocgtools@HideMiniLayer[\ocgt@@ls@jsclose]{#3}\vss}}}%
570 \fi
571 \@ocgtools@insidemini@layerfalse
572 }
573
574 </package>

```