The ocg-p package[*]

DI Werner Moshammer[†]

2013/01/10

Contents

Abstract

The ocg-p package provides the environment that allows to insert OCG (Optional Content Group) into PDF documents without JavaScript. These OCGs can be simply described as layers from the user's point of view.

The ocg-p package is intended as a full replacement for the file `ocg.sty` which is part of the asymptote package. While `ocg.sty` is limited to be used with `pdfLaTeX`, the ocg-p package can also be used with `XeLaTeX`. Additionally nested OCGs (layers inside of another layer) are handled as such.

---

[*] This manual correspondends to ocg-p v0.4, dated 2013/01/10

[†] Contact me when you find mistakes in the manual: `sendmail.werner@gmail.com`

1 Introduction

The `ocg-p` package provides the environment that allows to insert OCG (Optional Content Group) into PDF documents without JavaScript. These OCGs can be simply described as layers from the user's point of view.

OCGs are part of the PDF specification since version 1.5 and are described in the PDF Reference as:

> Optional content (PDF 1.5) refers to sections of content in a PDF document that can be selectively viewed or hidden by document authors or consumers. This capability is useful in items such as CAD drawings, layered artwork, maps, and multi-language documents.

OCGs are not part of the ISO standard 19005 `PDF/A-1`, but part of the newer `PDF/A-2` standard.

The `ocg-p` package is intended as a full replacement for the file `ocg.sty` which is part of the `asymptote` package. While `ocg.sty` is limited to be used with `pdfLaTeX`, the `ocg-p` package can also be used with `XeLaTeX`. Additionally nested OCGs (layers inside of another layer) are handled as such.

2 Usage

Here is a quick summary of the usage of `ocg-p`. The package consists of one main environment to create layers and a few commands for buttons to change the visibility of the layers in some way. Based on the ocg main environment there is an additionally environment available to create tables, which can be sorted by clicking on the headers.

2.1 Download

This package is available on CTAN[1]:

`CTAN: macros/latex/contrib/ocg-p/ocg-p.sty` The source file.

`CTAN: macros/latex/contrib/ocg-p/ocg-p.pdf` Documentation.

2.2 Package

Just load the package placing

```
\usepackage{ocg-p}
```

in the preamble of your LaTeX source file. There is only one option available for the package, which can be used to offer an additional environment to create tables which can be sorted by clicking on the headers. In this case load the package with the following line:

```
\usepackage[ocgtabular]{ocg-p}
```

---

[1] The Comprehensive TeX Archive Network `http://www.ctan.org/`

Important: If packages are used, which use the original `ocg` package then `ocg-p` should be loaded after these packages. The `ocg` environment from the `ocg` package is replaced in this case.

The `ocg` package is using the auxiliary file, so it is maybe necessary to compile your document 2 - 3 times until all layers are shown properly.

2.3 The ocg environment

This is the main environment of the `ocg-p` package. To create a OCG layer you have to use the `ocg` environment with three required arguments. Because it is intended as a replacement for the file `ocg.sty` this command can be used in the same way as it is used in `ocg.sty`.

```
\begin{ocg}{layer name}{layer id}{initial visibility}
content ...
\end{ocg}
```

The arguments are:

- *layer name*: This name is shown in in the layer toolbar of the (PDF) viewer, where the visiblity of the layers can be changed.
- *layer id*: A unique id which is internally used by the OCG environment to reference the layer. Only letters and numbers are allowed
- *initial visibility*: Sets the initial visibility when the document is opend. Only 0 and 1 are allowed (0 for invisible, 1 for visible)
- *content*: The content of the layer itself.

Beginning with `ocg-p` version 0.4 there are some optional options available to control the behaviour of the specified layer. Using this options the `ocg` environment is used the following way:

```
\begin{ocg}[opt1=val1, opt2=val2, ...]{layer name}{layer id}{initial visibility}
content ...
\end{ocg}
```

The options are given in a comma separated list of optionname value pairs. The usable options are:

`printocg`	This option can be set to the values `always`, `never` and `ifvisible`. The default value is `ifvisible`. It specifies the visibility state of the content in this layer when the document is printed. `ifvisible` means that the layer is printed only if it is visible in the document. `always` means that it is printed always, independent from the current visiblity state in the document, and `never` means that is never printed.
`exportocg`	This option can be set to the values `always`, `never` and `ifvisible`. The default value is `ifvisible`. It specifies the state for the content in this layer when the document is exported or saved to a format that does not support layers. `ifvisible` means that the layer is exported only if it is visible in the document. `always` means that it is exported always independent from the current visiblity state in the document, and `never` means that is never exported.
`listintoolbar`	This option can be set to the values `always`, `never` and `iffirstuse`. The default value is `iffirstuse`. `iffirstuse` means that the layer is only displayed in the toolbar when it is first inserted. `always` means that it is displayed every time this layer is inserted again, and `never` means that the layer is never displayed in the toolbar.

These options can be combined. So if you want for example that a layer can never be displayed in the document but is visible on a printing then you choose `listintoolbar=never`, `printocg=always` and a initial visible of `0` (invisible).

Important: Nested layers do not work with layers which are not visible in the layer toolbar of the browser.

2.4 The commands of the package

Beginning with `ocg-p` version 0.4 there are a few additional commands available. These commands can be used to add link actions (buttons) to the document, so that the visibility of some layers can be changed in some way. In all commands the `ocg/layer ids` should be given in a space separated list, and the last argument is for the link object itself. By default the link object is used by clicking with the mouse on it (mouseup event). But this behaviour can be changed by the optional options.

The command `toggleocg` can be used to toggle the visible of the given layers i the document.

```
\toggleocgs[optional options]{tlayerid1 tlayerid2 ...}{display}
```

The command `showocgs` can be used to make the given layers visible in the document.

```
\showocgs[optional options]{slayerid1 slayerid2 ...}{display}
```

The command `hideocgs` can be used to make the given layers invisible in the document.

```
\hideocgs[optional options]{hlayerid1 hlayerid2 ...}{display}
```

The command `setocgs` is a combination of the former commands. With `setocgs` it is possible to toggle some layers given as first argument list, to make some layers visible which are given in the second argument list and to make some layers invisible given in the third argument list.

```
\setocgs[optional options]{tlayerid1 tlayerid2 ...}
  {slayerid1 slayerid2 ...}{hlayerid1 hlayerid2 ...}{display}
```

There is one optional option availabe for all these commands, called `triggerocg`. These option can be set to the following values:

`onareaenter`	Using this value the action is performed by entering the link area with the cursor (a mousover effect).
`onareaexit`	Using this value the action is performed when the cursor exits the link area (a mouseout effect).
`onmousedown`	Using this value the action is performed when the mouse button is pressed inside the link area (a mousedown effect).
`onmouseup`	Using this value the action is performed when the mouse button isreleased inside the link area (a mouseup effect). `onmouseup` is the default value.

`allactions`	When this option is used all four trigger events can be used with different ocgs at the same time. To do this four lists of layer ids have to be given in the arguments of the commands. While one ocg layer id list is given as a space separated list, these lists should be comma separated.
	For example:
	`\toggleocgs[triggerocg=allactions]`
	`{aenid1 aenid2,aexid1,mdid1,muid1}[display}`
	The first list is for mouseover, the second for mouseout the third for mousedown and the last list for mouseup actions. Another example where only mouseenter and mousedown actions are used:
	`\toggleocgs[triggerocg=allactions]`
	`{aenid1 aenid2,,mdid1 mdid2}`

Important: It is not possible to make an action area inside of another action layer.

2.5 The ocgtabular environment

The purpose of the `ocgtabular` environment is to create tables which can be sorted by clicking on the headers. This environment should also show how the main `ocg` environment can be used to create another environment which can be useful.

To use the `ocgtabular` environmet the package option `ocgtabular` has to be used. In this way additional packages are imported which are necessary for this environment. The `ocgtabular` uses the original `tabular` package to create tables, where the data of the tables must be given in a database which is provided by the `datatool` package.

```
\begin{ocgtabular}[original tabular options]{original tabular argument}{ocgtabular options}
{datatool database name}
... original tabular/datatool code ...
\end{ocgtabular}
```

So the only difference to the `tabular` environment are the last two arguments where in the first the name of the database is given. This database is used to sort the data. The last argument is for additional options, but there are no options available at the moment.

For the header there is an addtional command available, but only inside of this environment.

```
\setocgtabularheader{columnname}{displayed header}
```

The `columnname` is the name of the column in the datatool database and `displayed header` is the text which is shown.

An example of this environment and its command is given in the following examples sections.

3 Examples

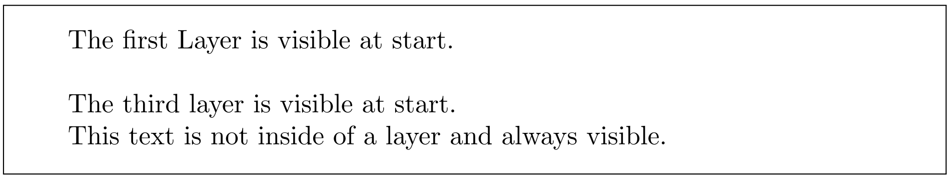
A few examples follow, showing the usage of this package.

3.1 Example 1: Three simple text layers

Here is a simple example with three layers with text content, where the second layer is set invisible when the document is opened. These commands are:

```
\documentclass{article}
\usepackage{ocg-p}
\begin{document}

\begin{ocg}{First layer}{oc1}{1}
  The first Layer is visible at start.
\end{ocg}

\begin{ocg}{Second layer}{oc2}{0}
  The second layer is not visible at start.
\end{ocg}

\begin{ocg}{Third layer}{oc3}{1}
 The third layer is visible at start.
\end{ocg}

This text is not inside of a layer and always visible.

\end{document}
```

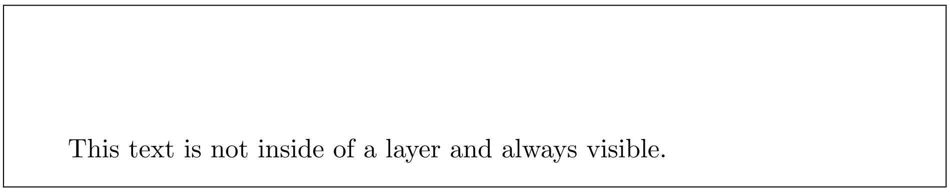
This will produce the following output, where the text in the second layer is invisible when the document is opend:



The first Layer is visible at start.

The third layer is visible at start.
This text is not inside of a layer and always visible.

The visiblity of the three layers can be changed with the layer toolbar of the viewer. If all layers are made invisible it looks like that:



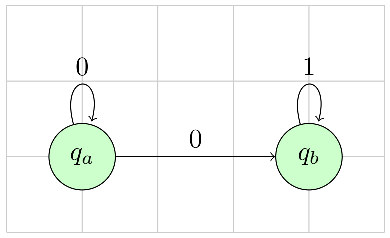
This text is not inside of a layer and always visible.

3.2 Example 2: OGC and the Ti*k*Z package

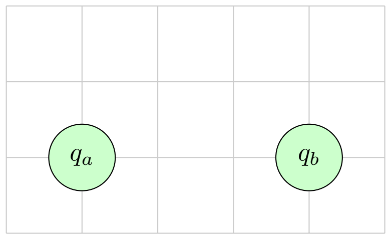
Using the ocg-p package with the Ti*k*Z package is very valuable, because it is possible to show or hide some parts of the picture with the layer toolbar of the viewer. Here a first code example:

```
\begin{tikzpicture}[node distance=3cm,every state/.style={fill=green!20},auto]
\begin{ocg}{grid}{ocgridid}{1}
  \draw[black!20] (-1,-1) grid (4,2);
\end{ocg}

\begin{ocg}{states}{ocstatesid}{1}
  \node[state] (q_a) {$q_a$};
  \node[state] (q_b) [right of=q_a] {$q_b$};
\end{ocg}

\begin{ocg}{edges}{ocedgesid}{1}
  \path[->]
  (q_a) edge node {0} (q_b)
        edge [loop above] node {0} ()
  (q_b) edge [loop above] node {1} ();
\end{ocg}
\end{tikzpicture}
```

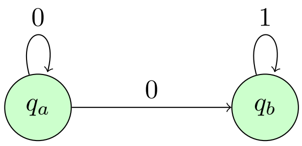
When the document is opend the following is shown:



But, for example, the edges could be made invisible with the layer toolbar:



Or the grid could be made invisible:



3.3 Example 3: OGC and the Ti*k*Z package again

These example should show how the layers can be used within text.

```
The following text can be toggled:
\begin{tikzpicture}[baseline=0]
\tikzstyle{nome}=[anchor=base,outer sep=0,inner sep=0,minimum height=.45cm,minimum
     width=4.4cm]
\begin{ocg}{blue text}{ocblueid}{1}
  \node[nome,blue]      (p1)  {\parbox[b][][t]{4.4cm}{This text is written in blue
       .}};
\end{ocg}
\begin{ocg}{red text}{ocredid}{0}
  \node[overlay,nome,red]  (p2)  {\parbox[b][][t]{4.4cm}{This text is written in red
       .}};
\end{ocg}
\end{tikzpicture}
And now the text is black again.
```

With the layer toolbar of the viewer it is possible to activate or deactivate the two layers, so there are three possibilities how it can be seen:

The following text can be toggled: This text is written in blue. And now the text is black again.

The following text can be toggled: This text is written in red. And now the text is black again.

The following text can be toggled: This text is written in red. And now the text is black again.

3.4 Example 4: OGC and link actions (buttons)

Combining the TikZ package with buttons to show, hide, toggle or set the layers adds much possibilities how this package can be used. Here is a way to show, how a table can be made, which can be sorted by clicking on the headers.

```
\usepackage{ocg-p}
\usepackage{tikz}        % will be needed for this example
\usepackage{datatool} % will be needed for this example
\usepackage{booktabs} % will be needed for this example

.
.
.

% generate database with data for the table
\DTLnewdb{sdata}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{John}
\DTLnewdbentry{sdata}{Lastname}{Doe}
\DTLnewdbentry{sdata}{Grade}{5}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Paul}
\DTLnewdbentry{sdata}{Lastname}{Bauer}
\DTLnewdbentry{sdata}{Grade}{1}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Peggy}
\DTLnewdbentry{sdata}{Lastname}{Sue}
\DTLnewdbentry{sdata}{Grade}{3}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Ever}
\DTLnewdbentry{sdata}{Lastname}{Last}
\DTLnewdbentry{sdata}{Grade}{4}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Werner}
\DTLnewdbentry{sdata}{Lastname}{Moshammer}
\DTLnewdbentry{sdata}{Grade}{1}

This table can be sorted by clicking on the headers:

\begin{tikzpicture}
\begin{ocg}{First Name}{ocfirstid}{0}
  \node[] (p1) {
\begin{tabular}{llc}
\toprule
\bfseries \setocgs{}{ocfirstid}{oclastid ocgradeid}{First name}
& \bfseries \setocgs{}{oclastid}{ocfirstid ocgradeid}{Last name}
& \bfseries \setocgs{}{ocgradeid}{ocfirstid oclastid}{Grade}
\DTLsort*{Firstname}{sdata}% sorted on the first name
\DTLforeach{sdata}{\first=Firstname, \last=Lastname,\grade=Grade}{%
\DTLiffirstrow{\\ \midrule}{\\}
\first & \last & \grade
}
\\ \bottomrule
\end{tabular}
};
\end{ocg}

\begin{ocg}{First Name}{oclastid}{1}
  \node[overlay] (p2) {
\begin{tabular}{llc}
\toprule
\bfseries \setocgs{}{ocfirstid}{oclastid ocgradeid}{First name}
& \bfseries \setocgs{}{oclastid}{ocfirstid ocgradeid}{Last name}
& \bfseries \setocgs{}{ocgradeid}{ocfirstid oclastid}{Grade}
\DTLsort*{Lastname}{sdata}% sorted on the last name
\DTLforeach{sdata}{\first=Firstname, \last=Lastname,\grade=Grade}{%
\DTLiffirstrow{\\ \midrule}{\\}
\first & \last & \grade
}
\\ \bottomrule
\end{tabular}
};
\end{ocg}

\begin{ocg}{First Name}{ocgradeid}{0}
  \node[overlay] (p3) {
\begin{tabular}{llc}
\toprule
\bfseries \setocgs{}{ocfirstid}{oclastid ocgradeid}{First name}
& \bfseries \setocgs{}{oclastid}{ocfirstid ocgradeid}{Last name}
& \bfseries \setocgs{}{ocgradeid}{ocfirstid oclastid}{Grade}
\DTLsort*{Grade}{sdata}% sorted on the grade
\DTLforeach{sdata}{\first=Firstname, \last=Lastname,\grade=Grade}{%
\DTLiffirstrow{\\ \midrule}{\\}
\first & \last & \grade
}
\\ \bottomrule
\end{tabular}
};
\end{ocg}

\end{tikzpicture}
```

The output is the following table:

This table can be sorted by clicking on the headers:

First name	**Last name**	**Grade**
Paul	Bauer	1
John	Doe	5
Ever	Last	4
Werner	Moshammer	1
Peggy	Sue	3

By clicking on `Grade` in the header the table changes the sorting and looks then as follows:

This table can be sorted by clicking on the headers:

First name	**Last name**	**Grade**
Paul	Bauer	1
Werner	Moshammer	1
Peggy	Sue	3
Ever	Last	4
John	Doe	5

3.5 Example 5: The ocgtabular environment

This example does the same as the last but now by using the ocgtabular environment.

```
\usepackage[ocgtabular]{ocg-p}
\usepackage{datatool} % will be needed for this example
\usepackage{booktabs} % will be needed for this example

.
.
.

% generate database with data for the table
\DTLnewdb{sdata}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{John}
\DTLnewdbentry{sdata}{Lastname}{Doe}
\DTLnewdbentry{sdata}{Grade}{5}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Paul}
\DTLnewdbentry{sdata}{Lastname}{Bauer}
\DTLnewdbentry{sdata}{Grade}{1}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Peggy}
\DTLnewdbentry{sdata}{Lastname}{Sue}
\DTLnewdbentry{sdata}{Grade}{3}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Ever}
\DTLnewdbentry{sdata}{Lastname}{Last}
\DTLnewdbentry{sdata}{Grade}{4}
\DTLnewrow{sdata}
\DTLnewdbentry{sdata}{Firstname}{Werner}
\DTLnewdbentry{sdata}{Lastname}{Moshammer}
\DTLnewdbentry{sdata}{Grade}{1}

This table can be sorted by clicking on the headers:

\begin{ocgtabular}{llc}{sdata}{}
\toprule%
\bfseries \setocgtabularheader{Firstname}{First name}
& \bfseries  \setocgtabularheader{Lastname}{Last name}
& \bfseries \setocgtabularheader{Grade}{Grade}
\DTLforeach{sdata}{\first=Firstname, \last=Lastname,\grade=Grade}{%
\DTLiffirstrow{\\ \midrule}{\\}
\first & \last & \grade
}
\\ \bottomrule%
\end{ocgtabular}
```

4 Possible future developement

These ideas may appear in new versions of the `ocg-p` package:

- The package should work with dvips. There is still something wrong at the moment.
- The package should use `.dtx` instead of `.sty`.
- Radio Button Groups (/RBGroups)

5 Implementation

The implementation is rather standard. At first main switches are defined to distinguish between the possible drivers `pdfLaTeX`, `XeLaTeX` and `dvips` (not fully implemented yet). Then the environment is defined.

```
% Copyright (C) 2012 by Werner Moshammer
% Parts of this code are Copyright (C) 2007 by Michael Ritzert <michael.ritz...
     @gmail.com>
% REPLACEMENT FOR THE OLD OCG.STY
% FOR PDFLATEX AND XELATEX (the old ocg.sys works only with pdflatex)
% This file may be distributed and/or modified under the LaTeX Project Public
     License

\NeedsTeXFormat{LaTeX2e}
\def\ocgpversion{0.4}
\ProvidesPackage{ocg-p}[2013/01/10 v\ocgpversion\space Optional Content Group in a
     PDF document]
% v0.1: 2012/11/01; v0.2: 2012/11/23; v0.3: 2012/12/01; v0.4: 2013/01/10
\RequirePackage{eso-pic}
\RequirePackage{ifpdf}
\RequirePackage{ifxetex}
\RequirePackage{xkeyval}

\newif\ifocgtabular
\DeclareOptionX[ocgp]{ocgtabular}{\ocgtabulartrue}
\DeclareOptionX*{\PackageWarning{ocg-p}{Option unknown: \CurrentOption}}
\ProcessOptionsX[ocgp]\relax

\ifocgtabular
  \RequirePackage{datatool}
  \RequirePackage{tikz}
  \RequirePackage{listings}
\fi

\newif\if@ocgp@ifps

\ifpdf
  \ifnum\pdftexversion<120
    \PackageError{ocg-p}{%
      pdfeTeX, version >= 1.20, required%
    }{%
       Install a newer version!%
    }%
  \fi
\else
  \ifxetex
     %already ok
  \else
    % dvips
    \@ocgp@ifpstrue
    \PackageWarningNoLine{ocg-p}{%
      Only XeLaTeX and pdfLaTeX are supported%
    }%
  \fi
\fi

\def\@ocgp@ocgHandle{\@auxout}
\newif\if@ocgp@iffirstrun\@ocgp@iffirstruntrue

\newif\if@ocgp@isnestedB\@ocgp@isnestedBfalse % nested OCG begin
\newif\if@ocgp@isnestedE\@ocgp@isnestedEfalse % nested OCG end
```

```
\def\@ocgp@nestedB{%
  \xdef\@ocgp@ocgorderlist{\@ocgp@ocgorderlist\space[}
}
\def\@ocgp@nestedE{%
  \xdef\@ocgp@ocgorderlist{\@ocgp@ocgorderlist\space]}
}


\providecommand\ocg[3]{} % if running with ocgtools
\renewenvironment{ocg}[4][]{%
  \if@ocgp@isnestedB% begin of nested ocg detected
    \immediate\write\@ocgp@ocgHandle{%
      \string\@ocgp@nestedB{}%
    }
  \fi
  \global\@ocgp@isnestedBtrue % ocg begin
  \global\@ocgp@isnestedEfalse % ocg end
  \if@filesw%
    \immediate\write\@ocgp@ocgHandle{%
      \string\@ocgp@newocg{#2}{#3}{#4}{#1}%
    }%
  \fi
  \gdef\@ocgp@curnum{#3}%
  \ifpdf
    \pdfliteral{/OC /OC\@ocgp@curnum\space BDC}%
  \else
    \if@ocgp@ifps % soon (not implemented yet) POSTSCRIPT
      \special{ps: mark /_objdef {psocgobj\@ocgp@curnum} /type/stream /OBJ pdfmark}
      \special{ps: mark {psocgobj\@ocgp@curnum} (/OC /OC\@ocgp@curnum\space BDC) /
           PUT pdfmark}
      %\special{ps: mark /OC /OC\@ocgp@curnum\space /BDC pdfmark}%
    \else
      \special{pdf: content /OC /OC\@ocgp@curnum\space BDC}%
    \fi
  \fi
  \message{/OC\@ocgp@curnum}%
  \ignorespaces
}{%
  \ifpdf
    \pdfliteral{EMC}%
  \else
    \if@ocgp@ifps % soon (not implemented yet) POSTSCRIPT
      %\special{ps: mark /EMC pdfmark}%
      \special{ps: mark {psocgobj\@ocgp@curnum} (EMC) /PUT pdfmark}
    \else
      \special{pdf: content EMC}%
    \fi
  \fi
  \if@ocgp@isnestedE% end of nested ocg detected
    \immediate\write\@ocgp@ocgHandle{%
      \string\@ocgp@nestedE{}%
    }
  \fi
  \global\@ocgp@isnestedEtrue % ocg end
  \global\@ocgp@isnestedBfalse % ocg begin
  \ignorespacesafterend
}


\def\@ocgp@ocglist{}
\def\@ocgp@ocgofflist{}
```

```
\def\@ocgp@ocgviewlist{} % to switch ocg off in layer toolbar of the viewer
\gdef\@ocgp@ocgmaplist{}
\def\@ocgp@ocgorderlist{} % ocgs in first-defined order + hierarchy

\define@choicekey*[ocgp]{ocg}{printocg}[\@ocgp@printbin\@ocgp@printno]{always,never
    ,ifvisible}[ifvisible]{%
  \ifcase\@ocgp@printno\relax
    \def\@ocgp@print{/Print<</PrintState /ON>>}%
  \or%
    \def\@ocgp@print{/Print<</PrintState /OFF>>}%
  \or%
    \def\@ocgp@print{}%
  \fi%
}
\define@choicekey*[ocgp]{ocg}{exportocg}[\@ocgp@exportbin\@ocgp@exportno]{always,
    never,ifvisible}[ifvisible]{%
  \ifcase\@ocgp@exportno\relax
    \def\@ocgp@export{/Export<</ExportState /ON>>}%
  \or%
    \def\@ocgp@export{/Export<</ExportState /OFF>>}%
  \or%
    \def\@ocgp@export{}%
  \fi%
}
\define@choicekey*[ocgp]{ocg}{listintoolbar}[\@ocgp@listbin\@ocgp@listno]{always,
    never,iffirstuse}[iffirstuse]{}

\define@choicekey*[ocgp]{ocgaction}{triggerocg}[\@ocgp@actionbin\@ocgp@actionno]{
    onareaenter,onareaexit,onmousedown,onmouseup,allactions}[onmouseup]{%
  \ifcase\@ocgp@actionno\relax
    \def\@ocgp@trigger{/E}%
  \or%
    \def\@ocgp@trigger{/X}%
  \or%
    \def\@ocgp@trigger{/D}%
  \or%
    \def\@ocgp@trigger{/U}%
  \or%
    \def\@ocgp@trigger{}%
  \fi%
}

\presetkeys[ocgp]{ocg}{printocg=ifvisible,exportocg=ifvisible,listintoolbar=
    iffirstuse}{}
\presetkeys[ocgp]{ocgaction}{triggerocg=onmouseup}{}


\newcount\@ocgp@num\@ocgp@num=0
\newcount\@ocgp@tonum\@ocgp@tonum=0
\newcount\@ocgp@sonum\@ocgp@sonum=0

\def\@ocgp@newocg#1#2#3#4{% #1: OCG name, #2: OC id num, #3: visiblity on/off
  \if@ocgp@iffirstrun
    \expandafter\ifx\csname OCG#2\endcsname\relax
      \expandafter\gdef\csname OCG#2\endcsname{#1}%
      \global\advance\@ocgp@num by 1
      \begingroup
        \setkeys[ocgp]{ocg}{#4}
        \ifpdf% PDFLATEX
          \immediate\pdfobj{<< /Type /OCG /Name (#1) /Usage <<%
            \@ocgp@print%
            %/View<</ViewState /OFF>> %
            \@ocgp@export%
            >> >>}% new ocg
          \xdef\@ocgp@curocg{\the\pdflastobj\space 0 R}% reference to current ocg id
        \else
          \if@ocgp@ifps % soon (not implemented yet) POSTSCRIPT
            \xdef\@ocgp@curocg{{@ocg\the\@ocgp@num}}% reference to current ocg id
            \special{ps: mark /_objdef \@ocgp@curocg /type/dict /OBJ pdfmark}
            \special{ps: mark \@ocgp@curocg << /Type /OCG /Name (#1)
              >> /PUT pdfmark}% new ocg
          \else % XELATEX
            \xdef\@ocgp@curocg{@ocg\the\@ocgp@num}% reference to current ocg id
            \special{pdf:obj \@ocgp@curocg\space <</Type/OCG /Name (#1) /Usage <<
              \@ocgp@print%
              %/View<</ViewState /OFF>> %
              \@ocgp@export%
              >> >>}% new ocg
          \fi
        \fi

        \expandafter\xdef\csname OCGpdfobj#2\endcsname{\@ocgp@curocg} % for ogcx-
             package
        \xdef\@ocgp@ocglist{\@ocgp@ocglist\space\@ocgp@curocg}% list of all OCGs in
             "first defined" order
        \ifnum\@ocgp@listno=1\else
          \xdef\@ocgp@ocgorderlist{\@ocgp@ocgorderlist\space\@ocgp@curocg} % all
               OCGs in "first defined" order + hierarchy
        \fi
      \endgroup
      \xdef\@ocgp@ocgmaplist{\@ocgp@ocgmaplist\space/OC#2\space\@ocgp@curocg\space
           ^^J} % name-to-id mapping
      \ifnum#3=1 %on
        % no list of all default-on OCGs needed, because of basestate on
      \else%
        \xdef\@ocgp@ocgofflist{\@ocgp@ocgofflist\space\@ocgp@curocg}% list of all
             default-off OCGs
      \fi%
    \else
      \message{OCG#2 reopened}% layer reopened
      \begingroup
        \setkeys[ocgp]{ocg}{#4}
        \ifnum\@ocgp@listno=0
          \xdef\@ocgp@ocgobjlist{}%
          \@ocgp@parseSpaceSeperatedList{#2}%
          \xdef\@ocgp@ocgorderlist{\@ocgp@ocgorderlist\space\@ocgp@ocgobjlist}
        \fi
      \endgroup
    \fi
  \fi
}


\AtBeginDocument{% AtEndDocument{% changed because of xetex problem in beamer class
  \@ocgp@iffirstrunfalse
  %\message{... \@ocgp@ocgorderlist ...}
  \ifpdf % PDFLATEX
    \pdfcatalog{%
      /OCProperties <<
        /OCGs [\@ocgp@ocglist]
        /D <</BaseState/ON /Order [\@ocgp@ocgorderlist] /OFF [\@ocgp@ocgofflist] /AS
              [%
          <</Event/View /OCGs [\@ocgp@ocglist] /Category[/View]>>%
          <</Event/Print /OCGs [\@ocgp@ocglist] /Category[/Print]>>%
          <</Event/Export /OCGs [\@ocgp@ocglist] /Category[/Export]>>%
        ]>>%
      >>%
    }
  \else
    \if@ocgp@ifps % soon (not implemented yet) POSTSCRIPT
      \special{ps: mark {Catalog} <<
        /OCProperties <<
          /OCGs [\@ocgp@ocglist]
          /D <</BaseState/ON /Order [\@ocgp@ocgorderlist] /OFF [\@ocgp@ocgofflist]>>
        >>
      >> /PUT pdfmark}%
    \else % XELATEX
      \special{pdf:put @catalog <<
        /OCProperties <<
          /OCGs [\@ocgp@ocglist]
          /D <</BaseState/ON /Order [\@ocgp@ocgorderlist] /OFF [\@ocgp@ocgofflist] /
               AS [%
            <</Event/View /OCGs [\@ocgp@ocglist] /Category[/View]>>%
            <</Event/Print /OCGs [\@ocgp@ocglist] /Category[/Print]>>%
            <</Event/Export /OCGs [\@ocgp@ocglist] /Category[/Export]>>%
          ]>>%
        >>%
      >>}%
    \fi
  \fi
}

\AtBeginDocument{%
  \ifpdf % PDFLATEX
    \immediate\pdfobj{<<\@ocgp@ocgmaplist\space>>}%
     \xdef\@ocgp@namesobj{\the\pdflastobj\space 0 R}%
     % append to pageresources
     \begingroup
     \edef\x{\endgroup
        \pdfpageresources{%
           \the\pdfpageresources
           /Properties \@ocgp@namesobj%
        }%
     }%
     \x
  \else
    \if@ocgp@ifps % soon (not implemented yet) POSTSCRIPT
      \AddToShipoutPicture{
        \special{ps: mark /_objdef {Resources} /type/stream /OBJ pdfmark}
        \special{ps: mark {Resources} << % it is something wrong here
        %  /Resources <<
          /Properties << ^^J%
          \@ocgp@ocgmaplist
        %>>
        >>
        >> /PUT pdfmark}}%
    \else  % XELATEX
      \AddToShipoutPicture{
        \special{pdf: put @resources <<
          /Properties << ^^J%
          \@ocgp@ocgmaplist
        >>
        >>}}%
    \fi
  \fi
}
```

```
% parsing a space-delimited ocgid-list to a space-delimited list of ocg-objects
\def\@ocgp@parseSpaceSeperatedList#1{\@ocgp@doparseSpaceSeperatedList#1 \relax}
\def\@ocgp@doparseSpaceSeperatedList#1 #2{%
  \ifcsname OCGpdfobj#1\endcsname%
    \xdef\@ocgp@ocgobjlist{\@ocgp@ocgobjlist\space\csname OCGpdfobj#1\endcsname}%
  \fi
  \ifx#2\relax
    %\@ocgp@ocgobjlist % only for debugging reasons
  \else
    \expandafter\@ocgp@doparseSpaceSeperatedList
  \fi
  #2%
}

%get the n-th element from a comma separated list
\newcommand\@ocg@selectElementN[2]{%
  \newcount\@ocgp@inum\@ocgp@inum=0%
  \def\@ocgp@tempN{}%
  \@for\@ocg@i:=#1\do{%
    \advance\@ocgp@inum by 1\relax
    \ifnum\@ocgp@inum=#2\relax
      \edef\@ocgp@tempN{\@ocg@i}%
    \fi
  }%
}

% link to toggle layers in a document without using the layer toolbar of the viewer
\newcommand\toggleocgs[3][]{%
  \setocgs[#1]{#2}{}{}{#3}%
}
% link to show layers in a document without using the layer toolbar of the viewer
\newcommand\showocgs[3][]{%
  \setocgs[#1]{}{#2}{}{#3}%
}
% link to hide layers in a document without using the layer toolbar of the viewer
\newcommand\hideocgs[3][]{%
  \setocgs[#1]{}{}{#2}{#3}%
}
% link to set layers in a document without using the layer toolbar of the viewer
\newcommand\setocgs[5][]{%
  \begingroup
    \setkeys[ocgp]{ocgaction}{#1}%
    \newcount\@ocgp@aanum\@ocgp@aanum=0
    \def\@ocgp@actionlist{}%
    \loop\ifnum\@ocgp@aanum<4
      \xdef\@ocgp@ocgobjlist{}%
      \advance\@ocgp@aanum by 1\relax
      \ifx\\#2\else%
        \expandafter\@ocg@selectElementN\expandafter{#2}{\@ocgp@aanum}%
        \ifx\@ocgp@tempN\@empty\else
          \xdef\@ocgp@ocgobjlist{/Toggle }%
          \expandafter\@ocgp@parseSpaceSeperatedList\expandafter{\@ocgp@tempN}%
        \fi
      \fi
      \ifx\\#3\else%
        \expandafter\@ocg@selectElementN\expandafter{#3}{\@ocgp@aanum}%
        \ifx\@ocgp@tempN\@empty\else
          \xdef\@ocgp@ocgobjlist{\@ocgp@ocgobjlist /ON }%
          \expandafter\@ocgp@parseSpaceSeperatedList\expandafter{\@ocgp@tempN}%
        \fi
      \fi
      \ifx\\#4\else%
        \expandafter\@ocg@selectElementN\expandafter{#4}{\@ocgp@aanum}%
        \ifx\@ocgp@tempN\@empty\else
          \xdef\@ocgp@ocgobjlist{\@ocgp@ocgobjlist /OFF }%
          \expandafter\@ocgp@parseSpaceSeperatedList\expandafter{\@ocgp@tempN}%
        \fi
      \fi
      \ifnum\@ocgp@actionno<4\relax% only one action
        \def\@ocgp@actionlist{\@ocgp@trigger << /S /SetOCGState /State [\@ocgp@ocgobjlist]>>}%
        \advance\@ocgp@aanum by 10\relax
      \else% allactions
        \ifcase\@ocgp@aanum\relax\or%
          \def\@ocgp@trigger{/E}%
        \or%
          \def\@ocgp@trigger{/X}%
        \or%
          \def\@ocgp@trigger{/D}%
        \or%
          \def\@ocgp@trigger{/U}%
        \fi
        \ifx\@ocgp@ocgobjlist\@empty\else
          \edef\@ocgp@actionlist{\@ocgp@actionlist\space\@ocgp@trigger << /S /SetOCGState /State [\@ocgp@ocgobjlist]>> }%
        \fi
      \fi
    \repeat
    \global\advance\@ocgp@sonum by 1\relax
    \leavevmode%
    \ifpdf%
      \pdfstartlink user {%
        %/Subtype /Link
        /Subtype /Widget
        /FT/Btn /Ff 65536
        /T(setocg\the\@ocgp@sonum)
        /H/N %Highlightning Mode: N=No;I=Invert;O=Outline;P=Push
        %/A <</S/SetOCGState /State [\@ocgp@ocgobjlist]>>
        /AA <<
           \@ocgp@actionlist
        >>
        %/Border [0 0 0]% no border
      }%
      #5\pdfendlink%
    \else%
      \if@ocgp@ifps % soon (not implemented yet) POSTSCRIPT
        \special{ps: bann
          << /Type /Annot
            %/Subtype /Link
            /Subtype /Widget
            /FT/Btn/Ff 65536
            /T(setocg\the\@ocgp@sonum)
            /H/N
            %/Border [0 0 0]% no border
            %/A <</S/SetOCGState /State [\@ocgp@ocgobjlist]>>%
            /AA <<
              \@ocgp@actionlist
            >>
          >>}#5%
        \special{ps:eann}%
      \else% XELATEX
        \special{pdf: bann
```

```
          << /Type /Annot
            %/Subtype /Link
            /Subtype /Widget
            /FT/Btn/Ff 65536
            /T(setocg\the\@ocgp@sonum)
            /H/N
            %/Border [0 0 0]% no border
            %/A <</S/SetOCGState /State [\@ocgp@ocgobjlist] >>%
            /AA <<
              \@ocgp@actionlist
            >>
          >>}#5%
        \special{pdf:eann}%
      \fi%
    \fi%
  \endgroup
}



% environment for a table which can be sorted by clicking on the header
% IMPLEMENTATION NOT FINISHED (last argument for options)
\ifocgtabular
  \lst@RequireAspects{writefile}
  \lstnewenvironment{ocgtabular}[4][]{
    \lstset{aboveskip=0pt,belowskip=0pt}
    \global\advance\@ocgp@tonum by 1\relax
    \newcommand\setocgtabularheader[2]{%
      \newcount\@ocgp@thnum\@ocgp@thnum=0%
      \xdef\@ocgp@ocgobjtlist{}%
      %\newcount\@ocgp@colindex%
      \DTLgetcolumnindex{\@ocgp@colindex}{#3}{##1}%
      \loop\ifnum\@ocgp@thnum<\DTLcolumncount{#3}%
      \advance\@ocgp@thnum by 1\relax
        \ifnum\@ocgp@thnum=\@ocgp@colindex\relax\else%
          \edef\@ocgp@ocgobjtlist{\@ocgp@ocgobjtlist\space ocgtabular\the\
               @ocgp@tonum o\the\@ocgp@thnum}%
        \fi%
      \repeat%
      \setocgs{}{ocgtabular\the\@ocgp@tonum o\@ocgp@colindex}{\@ocgp@ocgobjtlist
           }{##2}%
    }
    \newcount\@ocgp@tcnum\@ocgp@tcnum=0%
    \setbox\@tempboxa\hbox\bgroup
    \lst@BeginWriteFile{\jobname.oct}%
  }{
    \lst@EndWriteFile% closes output file
    \egroup
    %\vspace{-\baselineskip}%remove line from write-op
    \begin{tikzpicture}
    \loop\ifnum\@ocgp@tcnum<\DTLcolumncount{#3}
    \advance\@ocgp@tcnum by 1\relax
    \DTLgetkeyforcolumn{\@ocgp@theader}{#3}{\the\@ocgp@tcnum}
    \message{+++ocgtabular\the\@ocgp@tonum o\the\@ocgp@tcnum +++}
    \begin{ocg}[listintoolbar=never]{ocgtabular\the\@ocgp@tcnum}{ocgtabular\the\
         @ocgp@tonum o\the\@ocgp@tcnum}{\ifnum\@ocgp@tcnum=1 1\else 0\fi}
      \node[] (ocgnode\the\@ocgp@tcnum) {
        \DTLsort*{\@ocgp@theader}{#3}%
        \begin{tabular}[#1]{#2}
          \@@input\jobname.oct%
          %\input{\jobname.oct}%problems with some commands
        \end{tabular}
      };
    \end{ocg}
    \repeat
    \end{tikzpicture}
   }
 \fi
```

6 Change history

0.1 Initial version, only usable with `XeLaTeX`, not based on the `ocg` package and therefore the arguments were a little bit different.

0.2 The `ocg-p` package was made compatible with the `ocg` package using the same environment name and arguments and using the `aux` file in the same way. Support for `pdfLaTeX` was added. First public version.

0.3 Fixed bug in `ocg` environment (missing `\fi`). Fixed a bug if `XeLaTeX` is used. Nestes OCGs (layers inside another layer) are now handled as such.

0.4 Removed unnecessary `\makeatletter` and `\makeatother` commands, fixed an issue with the `beamer` class under `XeLaTeX` and other minor bugfixes. New options in the `ocg` environment. New commands for actions. New `ocgtabular` environment.

0.5 Planned: Not in a `.sty` file anymore, now using `.dtx`.