

The `ocg-p` package*

DI Werner Moshhammer†

2012/12/01

Contents

1	Introduction	2
2	Usage	2
2.1	Package	2
2.2	Commands and Environments	2
3	Examples	3
3.1	Example 1: Three simple text layers	3
3.2	Example 2: OGC and the <code>TikZ</code> package	4
3.3	Example 3: OGC and the <code>TikZ</code> package again	5
4	Possible future developement	5
5	Implementation	6
6	Change history	10

Abstract

The `ocg-p` package provides the environment that allows to insert OGC (Optional Content Group) into PDF documents without JavaScript. These OGCs 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 OGCs (layers inside of another layer) are handled as such.

*This manual correspondends to `ocg-p` v0.3, dated 2012/12/01

†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.

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 `pdf latex` , the `ocg-p` package can also be used with `x e latex`. Additionally nested OCGs are handled as such.

2 Usage

Here is a quick summary of the usage of `ocg-p`.

2.1 Package

Just load the package placing

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

in the preamble of your $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ source file. There are no options available at the moment.

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

2.2 Commands and Environments

At the moment the package has only one usable environment. 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 visibility 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 opened. Only 0 and 1 are allowed (0 for invisible, 1 for visible)
- *content*: The content of the layer itself.

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.

```
\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 opened:

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 visibility 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 TikZ package

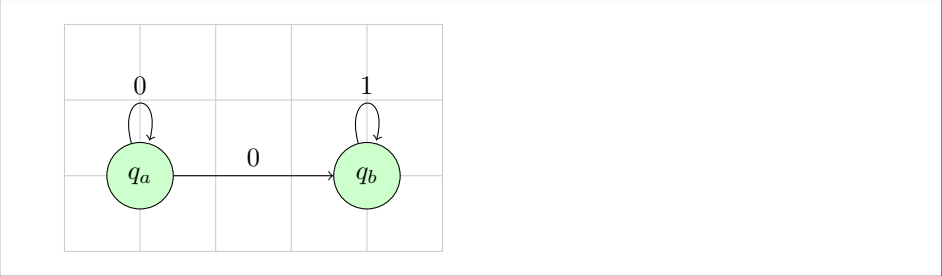
Using the `ocg-p` package with the `TikZ` package is very valuable, because it is possible to show or hide some parts of the picture with the layer toolbar of the viewer. Here a first code example:

```
\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}

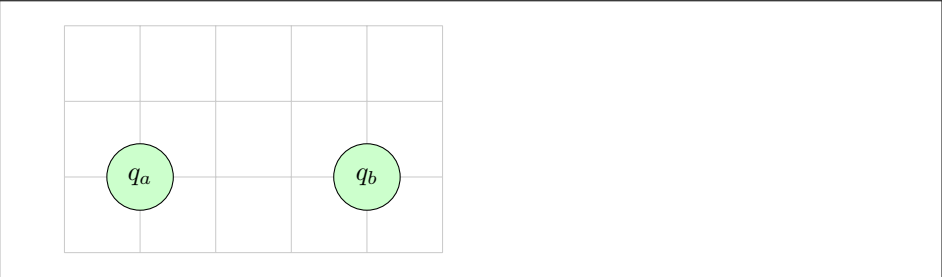
\begin{ocg}{states}{ocstatesid}{1}
  \node[state] (q_a) {$q_a$};
  \node[state] (q_b) [right of=q_a] {$q_b$};
\end{ocg}

\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}
```

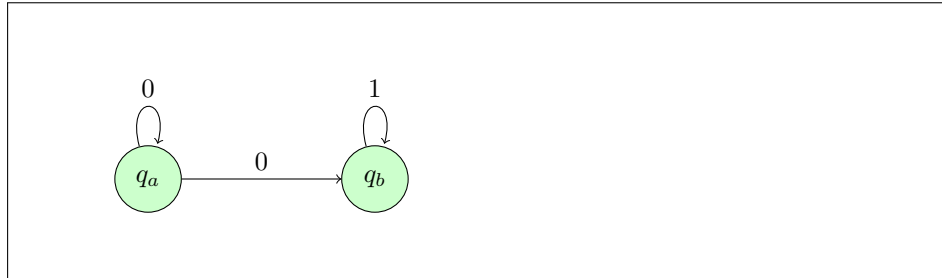
When the document is opened the following is shown:



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 TikZ 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 blue. And now the text is black again.

4 Possible future development

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

- The package should work with dvips.
- The ocg environment should accept options like notprintable, alwaysprinted, notinlayertoolbar (means that the visibility is not changeable), notexportable,

`alwaysexported`. So it is for example possible that a layer is not visible but is printed.

- Commands for buttons which make all layers visible or invisible.

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 Moshhammer
% 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{0.3}
\ProvidesPackage{ocg-p}[2012/12/01 v\ocgpversion\space Optional Content Group in a
  PDF document]

\RequirePackage{eso-pic}
\RequirePackage{ifpdf}
\RequirePackage{ifxetex}

\DeclareOption*{\ProcessOptions*\relax %allow anything as option for the moment}

\newif\if@ocgp@ifps

\ifpdf
  \ifnum\pdftexversion<120
    \PackageError{ocg-p}{%
      pdfTeX, 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@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 /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}%
    \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

% FUTURE PARAMETERS
% notprintable -> PrintState=OFF

```

```

% alwaysprinted -> Printstate=ON
% notinlayertoolbar (visibility not changeable) -> not listed in /Order
% notexportable -> ExportState=OFF
% alwaysexported -> Exportstate=ON
% visible (same as bool-argument visible) -> ViewState=ON
% not visible (same as bool-argument visible) -> ViewState=ON

% ->
% not visible AND alwaysprinted -> only on print

\newcount\ocgp@num\ocgp@num=0

\def\ocgp@newocg#1#2#3#4{% #1: OCG name, #2: OC id num, #3: visibility on/off
\expandafter\ifx\csname OCG#2\endcsname\relax
\expandafter\gdef\csname OCG#2\endcsname{#1}%
\global\advance\ocgp@num by 1
\ifpdf% PDFLATEX
\immediate\pdfobj{<< /Type /OCG /Name (#1) /Usage <<%
%/Print<</PrintState /ON>> %
%/View<</ViewState /OFF>> %
%/Export<</ExportState /OFF>>%
>> >>}% new ocg
\edef\ocgp@curocg{\the\pdfobj\space 0 R}% reference to current ocg id
\else
\ifocgp@ifps % soon (not implemented yet) POSTSCRIPT
\edef\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
\edef\ocgp@curocg{\ocg\the\ocgp@num}% reference to current ocg id
\special{pdf:obj \ocgp@curocg\space <</Type/OCG /Name (#1) /Usage <<
%/Print<</PrintState /ON>> %
%/View<</ViewState /OFF>> %
%/Export<</ExportState /OFF>>%
>> >>}% new ocg
\fi
\fi
\expandafter\edef\csname OCGpdfobj#2\endcsname{\ocgp@curocg} % for ogcx-package
\edef\ocgp@ocglist{\ocgp@ocglist\space\ocgp@curocg}% list of all OCGs in "
first defined" order
\edef\ocgp@ocgorderlist{\ocgp@ocgorderlist\space\ocgp@curocg} %
all OCGs in "first defined" order + hierarchy
\edef\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%
\edef\ocgp@ocgofflist{\ocgp@ocgofflist\space\ocgp@curocg}% list of all
default-off OCGs
\fi%
\else
% layer reopened
\fi
}

\AtEndDocument{%
%\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
\beginingroup
\edef\x{\endgroup
\pdfpageresources{%
\the\pdfpageresources
/Properties \@ocgp@namesobj%
}%
}%
\x
\else
\if@ocgp@ifps % soon (not implemented yet) POSTSCRIPT
%\AddToShipoutPicture{
%\special{ps: mark {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
\makeatletter
\def\@ocgp@parseSpaceSeperatedList#1{ \@ocgp@doparseSpaceSeperatedList#1 \@empty }
\def\@ocgp@doparseSpaceSeperatedList#1 #2{%
  \ifcsname OCGpdfobj#1\endcsname
    \xdef\@ocgp@ocgobjlist{\@ocgp@ocgobjlist\space\csname OCGpdfobj#1\endcsname}
  \fi
  \ifx #2\@empty
    %\@ocgp@ocgobjlist % only for debugging reasons
  \else
    \expandafter\@ocgp@doparseSpaceSeperatedList
  \fi
  #2%
}
\makeatother

% link to toggle a layer in a document without using the layer toolbar of the viewer
\makeatletter
\long\def\toggleocg#1#2{%
  \xdef\@ocgp@ocgobjlist{}%
  \@ocgp@parseSpaceSeperatedList{#1}
  \leavevmode%
  \ifpdf
    \pdfstartlink user {%
      /Subtype /Link
      /A << /S/SetOCGState /State [/Toggle \@ocgp@ocgobjlist] >>
      %/A << /Type /Action /S /GoTo /D [1 0 R] >>
      /Border [0 0 0]% no border
    }%
    #2\pdfendlink%
  \else
    \special{pdf: bann
      << /Type /Annot
        /Subtype /Link
        /Border [0 0 0]% no border
        /A <</S/SetOCGState /State [/Toggle \@ocgp@ocgobjlist] >>
      >>}#2%
    \special{pdf:eann}
  \fi
}
\makeatother

```

6 Change history

- 0.1 Initial version, only usable with xetex, 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`). Nestes OCGs (layers inside another layer) are now handled as such.
- 0.4 Planned: Not in a `.sty` file anymore, now using `.dtx`.