The **pdflscape** package

Heiko Oberdiek

&lt;heiko.oberdiek at googlemail.com&gt;

2008/08/11 v0.10

Abstract

Package **pdflscape** adds PDF support to the environment **landscape** of package **lscape** by setting the PDF page attribute `/Rotate`.

Contents

1 Documentation 2
   1.1 Supported drivers ........................................... 2
   1.2 Caveat ......................................................... 2
   1.3 Requirements ................................................. 2
   1.4 Usage ........................................................ 2

2 Implementation 2
   2.1 Package identification ...................................... 2
   2.2 Driver options ................................................. 2
   2.3 Autodetection of driver ..................................... 3
      2.3.1 Driver **pdftex** ....................................... 3
      2.3.2 Driver **xetex** ........................................ 3
      2.3.3 Detect driver based on \texttt{\Gin@driver} ............ 4
      2.3.4 Driver **dvips** ....................................... 4
   2.4 Driver implementation ...................................... 4
      2.4.1 **pdflatex** ............................................ 5
      2.4.2 PostScript driver ....................................... 5
      2.4.3 Driver **dvipdfm** ..................................... 6
   2.5 Driver independent stuff .................................. 6

3 Test 7
   3.1 Driver detection tests .................................... 7
   3.2 Test for \texttt{\PLS@CheckAngle} ......................... 7
   3.3 Test for rotate ............................................. 8

4 Installation 10
   4.1 Download .................................................... 10
   4.2 Bundle installation ....................................... 10
   4.3 Package installation ..................................... 10
   4.4 Refresh file name databases .............................. 11
   4.5 Some details for the interested ........................ 11

5 Catalogue 12
1 Documentation

1.1 Supported drivers

- pdftex
- dvips, dvipson, pctex32, pctexps (and other drivers that provide a non-empty \Gin@PS@raw)
- dvipdfm

1.2 Caveat

Depending on the configuration Ghostscript adds a guessed rotation entry by its own. This can lead to two /Rotate entries per page. To prevent this behaviour set the parameter AutoRotatePages to /None, eg:

```
ps2pdf -dAutoRotatePages=/None
```

1.3 Requirements

- The package l scape.
- The package atbegshi for all drivers except pdftex.

1.4 Usage

Load this package instead of or after package lscape:

```
\usepackage{pdflscape}
\begin{landscape}...
```

2 Implementation

2.1 Package identification

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{pdflscape}[2008/08/11 Display of landscape pages in PDF (HO)]
```

2.2 Driver options

```
\let\PLS@option\@empty
\let\PLS@driver\@empty
```
2.3 Autodetection of driver

2.3.1 Driver pdftex

\RequirePackage{ifpdf}[2006/02/20]
\ifpdf
  \def\PLS@temp{pdftex}
  \ifpdf
    \PackageInfo{pdflscape}{Auto-detected driver: \PLS@temp}
  \fi
\fi

2.3.2 Driver xetex

\RequirePackage{ifxetex}
2.3.3 Detect driver based on \Gin@driver

2.3.4 Driver dvips

2.4 Driver implementation

• \PLS@AddRotate #1
  it expects the correct rotation number in #1 and implements the adding of the /Rotation entry in the /Page object of the current page.

• \PLS@RemoveRotate
  it removes a previous /Rotate entry, if necessary.
2.4.1 pdfTeX

Not too nice is the global setting of `\pdfpageattr`. Perhaps this can be changed in future versions.

```latex
\def\PLS@temp{pdftex}
\ifx\PLS@temp\PLS@driver
\def\PLS@AddRotate#1{%
\ifnum#1=0 %
Already the default.
\else
\global\pdfpageattr\expandafter{%
\the\pdfpageattr
/Rotate #1%
}%
\fi
}\fi

Removes a /Rotate entry.
\def\PLS@RemoveRotate{%
\begingroup
\global\pdfpageattr\expandafter{\expandafter}
\expandafter\PLS@@RemoveRotate
\the\pdfpageattr /Rotate@nil
\endgroup
}%
\def\PLS@@RemoveRotate#1/Rotate#2@nil{%
Append /Rotate free stuff to \pdfpageattr.
\global\pdfpageattr\expandafter{\the\pdfpageattr#1}%
\ifx\,#2\%
Ready, because the detected /Rotate is part of the end marker: /Rotate@nil
\else
First read in the argument of /Rotate, then continue parsing.
\afterassignment\PLS@RemoveRotate
\count0=#2@nil
\fi
}\fi
```

2.4.2 PostScript driver

```latex
\def\PLS@temp{dvips}
\ifx\PLS@temp\PLS@driver
\RequirePackage{atbegshi}%
\AtBeginShipout{\PLS@AtBeginShipout}%
\let\PLS@AtBeginShipout\@empty
\def\PLS@AddRotate#1{%
\ifnum#1=0 %
\else
\def\PLS@AtBeginShipout{%
\global\setbox\AtBeginShipoutBox\vbox{%
\Gin@PS@raw{%
/{{ThisPage}\string<\string</Rotate #1\string>\string}>% 
/PUT pdfmark%
}%
\box\AtBeginShipoutBox
}\fi
}\fi
\def\PLS@RemoveRotate{\let\PLS@AtBeginShipout\@empty}%
```

5
2.4.3 Driver dvipdfm

\def\PLS@temp{dvipdfm}
\ifx\PLS@temp\PLS@driver
\RequirePackage{atbegshi}\
\AtBeginShipout{\PLS@AtBeginShipout}\
\let\PLS@AtBeginShipout\@empty
\def\PLS@AddRotate#1{\ifnum#1=0\else\def\PLS@AtBeginShipout{\special{pdf: put @thispage <<\Rotate #1>>}}\box\AtBeginShipoutBox\fi}
\def\PLS@RemoveRotate{\let\PLS@AtBeginShipout\@empty}
\fi

2.5 Driver independent stuff

The landscape environment is extended by adding the correct /Rotate entries.
\g@addto@macro{\landscape}{\PLS@Rotate{90}}
\g@addto@macro{\endlandscape}{\PLS@Rotate{0}}

\PLS@Rotate Main macro, that sets the /Rotate entry.
Argument: any TeX number or nothing, that means zero.
Driver independent.
\def\PLS@Rotate#1{\begingroup
1. Check and validate the argument.
\PLS@CheckAngle{#1}\expandafter\endgroup
2. Remove previous /Rotate entry.
\expandafter\PLS@RemoveRotate
3. Add /Rotate entry.
\expandafter\PLS@AddRotate\expandafter{\the\count@}
}

\PLS@CheckAngle Validates the rotation angle.
The result is stored in the count register \count0. Driver independent.
\def\PLS@CheckAngle#1{\ifx\#1\%\count@=0\else\count@=#1\relax\fi
Normalize to interval -360 < \count0 < 360.
\@whilenum\count@>359\do{\advance\count0 -360 }\@whilenum\count0<-359\do{\advance\count0 360 }\ifnum 1=0\ifnum\count@=0\else\ifnum\count@=90\else\ifnum\count@=-90\else
3 Test

3.1 Driver detection tests

\NeedsTeXFormat{LaTeX2e}
\documentclass{minimal}
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\usepackage{ifpdf}
\ifpdf
\def\ExpectDriver{pdftex}%
\fi
\usepackage{ifxetex}
\ifxetex
\def\ExpectDriver{dvipdfm}%
\fi
\usepackage{pdflscape}[2008/08/11]
\usepackage[pdftex]{pdflscape}[2008/08/11]
\usepackage[dvipdfmx]{pdflscape}[2008/08/11]
\def\ExpectDriver{dvips}
\usepackage[dvipsone]{graphics}
\usepackage{pdflscape}[2008/08/11]
\begin{document}
\begin{qstest}{driver}{driver}
\makeatletter
\Expect*{\PLS@driver}*{\ExpectDriver}%
\end{qstest}
\end{document}

3.2 Test for \PLS@CheckAngle

\begin{document}
\begin{qstest}{driver}{driver}
\makeatletter
\Expect*{\PLS@CheckAngle}{\ExpectDriver}%
\end{qstest}
\end{document}
\NeedsTeXFormat{LaTeX2e}
\documentclass{minimal}
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}
\usepackage{pdflscape}[2008/08/11]
\begin{document}
\begin{qstest}{checkangle}{checkangle}
\makeatletter
\def\Result{\
\def\ResultString{\the\count@}\
}
\def\PackageError#1#2#3{\
\def\Result{\
\def\ResultString{error}\
}
}
\def\Test#1#2{\
\begingroup\
\PLS@CheckAngle{#1}\
\Result\
\Expect*{\ResultString}{#2}\
\endgroup\
}
\Test{0}{0}\
\Test{90}{90}\
\Test{180}{180}\
\Test{270}{270}\
\Test{360}{0}\
\Test{450}{90}\
\Test{540}{180}\
\Test{630}{270}\
\Test{720}{0}\
\Test{3600}{0}\
\Test{3690}{90}\
\Test{-90}{270}\
\Test{-180}{180}\
\Test{-270}{-90}\
\Test{-360}{0}\
\Test{-450}{270}\
\Test{-540}{180}\
\Test{-630}{90}\
\Test{-720}{0}\
\Test{-3600}{0}\
\Test{-3690}{270}\
\Test{1}{error}\
\Test{-1}{error}\
\Test{123}{error}\
\end{qstest}
\end{document}
foo bar foo bar foo bar foo bar foo bar foo bar.
\end{landscape}
\section{Portrait Section}
Foo bar
\begin{landscape}
\section{Again Landscape Section}
\newpage
\section{Second Page of Landscape Section}
\end{landscape}
\section{Second Last Portrait Page}
\newpage
\section{Last Portrait Page}

---

```java
import java.io.FileInputStream;
import java.io.FileWriter;
import org.pdfbox.pdfparser.PDFParser;
import org.pdfbox.pdmodel.PDDocument;
import org.pdfbox.pdmodel.PDPage;

public class ExtractRotate {
    public static void main(String[] args) {
        try {
            String infile = args[0];
            String outfile = args[1];
            FileWriter out = new FileWriter(outfile);
            PDFParser parser =
                new PDFParser(new FileInputStream(infile));
            parser.parse();
            PDDocument document = parser.getPDDocument();
            PDDocumentCatalog catalog = document.getDocumentCatalog();
            int i = 0;
            for (Object page: catalog.getAllPages()) {
                i++;
                out.write("/Page " + i + " " + "/Rotate "
                         + ((PDPage)page).findRotation() + "\n");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

---

(*result6)*
/Page 1 /Rotate 0
/Page 2 /Rotate 90
/Page 3 /Rotate 0
/Page 4 /Rotate 90
/Page 5 /Rotate 90
/Page 6 /Rotate 0
/Page 7 /Rotate 0
(*result6)*

/**
 * ExtractRotate.java
 *
 * Copyright (C) 2007 by Heiko Oberdiek <heiko.oberdiek at googlemail.com>
 *
 * Requires: PDFBox (http://www.pdfbox.org/)
 *
 * Syntax: java ExtractRotate <pdffile> <textfile>
 *
 * The <pdffile> is analyzed and for each page its rotation
 * setting is printed in the <textfile>. Example:
 *
 * /Page 1 /Rotate 0
 * /Page 2 /Rotate 90
 *
 */

import java.io.FileInputStream;
import java.io.FileWriter;
import org.pdfbox.pdfparser.PDFParser;
import org.pdfbox.pdmodel.PDDocument;
import org.pdfbox.pdmodel.PDPage;

public class ExtractRotate {
    public static void main(String[] args) {
        try {
            String infile = args[0];
            String outfile = args[1];
            FileWriter out = new FileWriter(outfile);
            PDFParser parser =
                new PDFParser(new FileInputStream(infile));
            parser.parse();
            PDDocument document = parser.getPDDocument();
            PDDocumentCatalog catalog = document.getDocumentCatalog();
            int i = 0;
            for (Object page: catalog.getAllPages()) {
                i++;
                out.write("/Page " + i + " " + "/Rotate "
                         + ((PDPage)page).findRotation() + "\n");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
4 Installation

4.1 Download

Package. This package is available on CTAN:\footnote{ftp://ftp.ctan.org/tex-archive/}:


Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard “A Directory Structure for \TeX\ Files” (CTAN:tds/tds.pdf). Directories with \texttt{texmf} in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the oberdiek.tds.zip in the TDS tree (also known as texmf tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory TDS:scripts/oberdiek/ for scripts that need further installation steps. Package attachfile2 comes with the Perl script \texttt{pdfatfi.pl} that should be installed in such a way that it can be called as \texttt{pdfatfi}. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

Unpacking. The .dtx file is a self-extracting docstrip archive. The files are extracted by running the .dtx through plain \TeX: 

```
tex pdflscape.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as texmf tree):
4.4 Refresh file name databases

If your \TeX\ distribution (\TeX\, m\TeX, …) relies on file name databases, you must refresh these. For example, \TeX\ users run \texttt{texhash} or \texttt{mktexlsr}.

4.5 Some details for the interested

\textbf{Attached source.} The PDF documentation on CTAN also includes the \texttt{.dtx} source file. It can be extracted by AcrobatReader 6 or higher. Another option is \texttt{pdftk}, e.g. unpack the file into the current directory:

\begin{verbatim}
pdftk pdflscape.pdf unpack_files output .
\end{verbatim}

**Unpacking with LaTeX.** The \texttt{.dtx} chooses its action depending on the format:

- **plain \TeX:** Run \texttt{docstrip} and extract the files.
- **\LaTeX:** Generate the documentation.

If you insist on using \LaTeX\ for \texttt{docstrip} (really, \texttt{docstrip} does not need \LaTeX), then inform the autodetect routine about your intention:

\begin{verbatim}
latex \let\install=y\input{pdflscape.dtx}
\end{verbatim}

Do not forget to quote the argument according to the demands of your shell.

**Generating the documentation.** You can use both the \texttt{.dtx} or the \texttt{.drv} to generate the documentation. The process can be configured by the configuration file \texttt{ltxdoc.cfg}. For instance, put this line into this file, if you want to have A4 as paper format:

\begin{verbatim}
\PassOptionsToClass{a4paper}{article}
\end{verbatim}

An example follows how to generate the documentation with \texttt{pdflatex}:

\begin{verbatim}
pdflatex pdflscape.dtx
makeindex -s gind.ist pdflscape.idx
pdflatex pdflscape.dtx
makeindex -s gind.ist pdflscape.idx
pdflatex pdflscape.dtx
\end{verbatim}
5 Catalogue

The following XML file can be used as source for the \TeX\ Catalogue. The elements \texttt{caption} and \texttt{description} are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is \texttt{pdflscape.xml}.

\begin{verbatim}
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='pdflscape'>
  <name>pdflscape</name>
  <caption>Make landscape pages display as landscape.</caption>
  <authorref id='auth:oberdiek'/>
  <license type='lppl1.3'/>
  <version number='0.10'/>
  <description>
    The package adds PDF support to the landscape environment of \texttt{lscape}, by setting the PDF \texttt{/Rotate} page attribute. Pages with this attribute will be displayed in landscape orientation by conforming PDF viewers.
  </description>
  <documentation details='Package documentation' href='ctan:/macros/latex/contrib/oberdiek/pdflscape.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/pdflscape.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
\end{verbatim}

6 History

[2001/01/15 v0.1]
- First public version, published in \texttt{de.comp.text.tex}:
  "Re: Querformat, pdfLaTex und thumbpdf"\footnote{Url: \url{http://groups.google.com/group/de.comp.text.tex/msg/e054c5795e52d2b8}}

[2001/02/04 v0.2]
- Minor documentation update.
- CTAN.

[2004/05/11 v0.3]
- Support for dvipdfm added.

[2004/05/12 v0.4]
- Bug fix: support for multipage landscape environment for all drivers except \texttt{pdftex}.

[2006/02/20 v0.5]
- DTX framework.
- LPPL 1.3
• Code is not changed.

[2006/04/24 v0.6]
• Bug fix: \PLS@RemoveRotate reinserted for \PLS@Rotate.
• Fix for \PLS@RemoveRotate (dvips/dvipdfm): \PLS@EveryShi hook is cleared.

[2007/04/11 v0.7]
• Line ends sanitized.

[2007/04/17 v0.8]
• Package atbegshi replaces everyshi.

[2007/10/21 v0.9]
• Driver detection for XƎTEX added.
• Fix for rotation angles \leq -360 or \geq 360.

[2008/08/11 v0.10]
• Code is not changed.
• URLs updated.

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

Symbols
\@ehc .......................... 50, 68, 112, 207
\@empty . 5, 6, 20, 22, 29, 33, 41, 57, 73, 90, 96, 109, 145, 160, 166, 178
\@nil .......................... 129, 132, 137
\@undefined ...................... 92
\@whilenum ..................... 195, 196
\ \ .............................. 134, 190

A
\advance .......................... 195, 196
\afterassignment ................. 136
\AtBeginShipout ................. 144, 165
\AtBeginShipoutBox 150, 155, 171, 173

B
\begin . 244, 245, 258, 259, 307, 309, 316
\box .......................... 155, 173

C
\count .......................... 137
\count@ .......................... 187, 191, 193, 195, 196, 197, 198, 199, 200, 201, 202, 203, 206, 262
\CurrentOption .......................... 9, 23, 25

D
\DeclareOption .......................... 8, 21
\do .......................... 195, 196
\documentclass ..................... 214, 253, 305

E
\end . 248, 249, 300, 301, 313, 320, 324
\endlandscape ..................... 181
\Expect .......................... 247, 273
\ExpectDriver .......................... 222, 226, 231, 235, 239, 247

G
\g@addto@macro .......................... 180, 181
\Gin@driver .......................... 77
\Gin@PS@raw ......................... 92, 94, 96, 151

I
\ifnum ......................... 92, 117, 147, 168, 197, 198, 199, 200, 201, 202, 203
\ifpdf ........................ 40, 221
\ifix ......................... 22, 29, 33, 41, 45, 57, 63, 73, 77, 90, 92, 94, 96, 109, 115, 134, 142, 163, 190
\ifxetex .......................... 56, 225
\IncludeTests ..................... 216, 255
L
\landscape ........................ 180
\LogTests .......................... 217, 256

M
\makeatletter .................. 246, 260
\MessageBreak ..................... 48, 66

N
\n .................................. 373
\NeedsTeXFormat ............. 2, 213, 252, 304
\newpage ............................ 318, 322

P
\PackageError ............ 47, 65, 110, 205, 264
\PackageInfo ............ 43, 59, 79, 104
\PassOptionsToPackage .... 31, 35
\pdfpageattr ............. 119, 120, 127, 129, 133
\PLS@@RemoveRotate .... 128, 132, 136
\PLS@AddRotate ........ 116, 146, 167, 187
\PLS@AtBeginShipout ...... 144,
  145, 149, 160, 165, 166, 170, 178
\PLS@CheckAngle .......... 184, 189, 271
\PLS@driver ........ 6, 10, 41,
  42, 45, 51, 57, 58, 63, 69, 73, 78,
  80, 90, 103, 109, 115, 142, 163, 247
\PLS@option ........... 5, 9, 29, 31, 48, 66
\PLS@RemoveRotate ... 125, 160, 178, 186
\PLS@Rotate ........ 180, 181, 182
\PLS@temp 7, 13, 14, 15, 16, 17, 18, 19,
  20, 22, 23, 25, 33, 35, 39, 42, 43,
  45, 51, 55, 58, 60, 63, 69, 72, 75,
  77, 85, 86, 87, 88, 89, 91, 103,
  105, 114, 115, 141, 142, 162, 163
\ProcessOptions ............... 28
\ProvidesPackage ............ 3

R
\RequirePackage .. 37, 38, 54, 143, 164
\Result ............................. 261, 265, 272
\ResultString ............. 262, 266, 273

S
\section 308, 310, 314, 317, 319, 321, 323
\setbox ............................. 150, 171
\space .................. 60, 80, 102, 105, 206
\special ............................ 172

T
\Test ........................... 269, 276, 277, 278, 279, 280,
  281, 282, 283, 284, 285, 286,
  287, 288, 289, 290, 291, 292,
  293, 294, 295, 296, 297, 298, 299
\the ............................ 120, 129, 133, 187, 206, 262

U
\usepackage .... 215, 220, 224, 228,

V
\vbox .............................. 150, 171