A LaTeX Style for Typesetting a Three-Dimensional Product Box*

Gerd Neugebauer
Im Lerchelsböhl 5
64521 Groß-Gerau (Germany)
Email: gene@gerd-neugebauer.de

Documentation date: 2010/12/29

Abstract

The package `productbox` provides a style file for typesetting a three-dimensional product box. This product box can be rendered as it is standing on a surface and some light is shed onto it. Alternatively it can be typeset as a wireframe to be cut out and glued together. This will lead to a physical product box.

*This file documents productbox.dtx version 1.1 (from revision 8333) as of 2010/12/29.

Contents

1 Introduction

Humans are tied to the physical world. Even in the age of software it is desirable to have a physical representation for it. In the commercial world the software is sold in product boxes containing a CD or DVD and printed documentation.

But even for open source software having a product box provides a means to associate a physical object with the more or less virtual entity "software".

There are many programs around devoted to producing layouts for product boxes. Each graphics program can be used for this purpose as well. Nevertheless the integration of material from the TEX world is not that easy.

The LATEX package `productbox` is an attempt to use the possibilities for LATEX and some packages to provide a means for typesetting the faces of a product box and assemble them into a three-dimensional image.

Note that this "image" is in fact a PDF object. It can be scaled without loss of quality[1]. It is also possible to extract the text from the product box by cut and paste in an appropriate PDF reader.

2 The User Interface

The LATEX style `productbox` is based on Ti*k*Z at [Tan08] least in version 2.0. It is best used with a LATEX variant which is able to produce PDF. If no proper tool chain is used then some of the effects should not be used.

ProductBox

The environment `ProductBox` provides the central means for producing a product box. The contents is used to define the appearance of the faces. Finally the selected type of output is produced.

The content of the environment consists mainly of the definition of the six faces. Not all of them need to be defined. If one face is not defined then it appears as an empty rectangle of white color.

Thus you usually want to define the faces. Two approaches are provided to define the content of the faces. The simplified interface does not really require any knowledge of the underlying Ti*k*Z package. Just some knowledge of LATEX is sufficient. The extended interface opens the full power to the user. Both interface types can be freely intermixed. The tow kinds of interfaces are described in section 2.1 and 2.2.

The environment `ProductBox` can be controlled with a number of optional parameters. Those parameters are described in section 2.3.

[1] unless some pixel images have been included

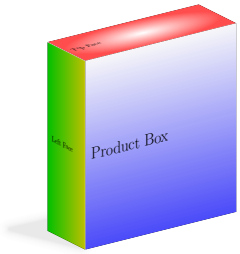
2.1 The Simplified User Interface

The simplified user interface is meant for someone not familiar with Ti*k*Z. It encapsulates nearly anything and relies on just some basic LaTeX experience. As a downside the functionality is restricted. Any fancy artwork on the box background can hardly be achieved.

The following example shows a complete – even rather useless – definition of a product box with the simplified user interface.

```
\begin{ProductBox}
  \begin{FrontFace}[bottom color=white!30!blue,top color=white]
    \Huge Product Box
  \end{FrontFace}
  \begin{BackFace}[top color=yellow!30!red,bottom color=white]
    \large Back Face
  \end{BackFace}
  \begin{TopFace}[outer color=white!30!red,inner color=white]
    \large Top Face
  \end{TopFace}
  \begin{BottomFace}[outer color=red,inner color=black]
    \large Bottom Face
  \end{BottomFace}
  \begin{LeftFace}[left color=green,right color=yellow]
    \large Left Face
  \end{LeftFace}
  \begin{RightFace}[left color=white,right color=black]
    \large Right Face
  \end{RightFace}
\end{ProductBox}
```

This code renders as



The content of the environment `ProductBox` is simply evaluated. It can contain any code you like. Useful for the production of a product box are some inner environments. They are called `FrontFace`, `BackFace`, `LeftFace`, `RightFace`, `TopFace`, and `BottomFace`. They are defined inside the main environment only. Those environments can be used to specify the contents of the respective faces of the box.

Note that in the three-dimensional rendering will show at most three of the faces. It does not hurt to define all of them, even if they are not shown at all.

FrontFace The environment `FrontFace` is used to define the content of the font face. The

environment processes its contents inside a `minipage` of the default width 88 mm reduced by the left and right separator width (`faceSep`).

The minipage is centered vertically on the face. Usually anything extending the default height of 100 mm is clipped.

```
\begin{FrontFace}
    ...
\end{FrontFace}
```

BackFace The environment `BackFace` is used to define the content of the back face. The environment processes its contents inside a `minipage` of the default width 88 mm reduced by the left and right separator width (`faceSep`).

The minipage is centered vertically on the face. Usually anything extending the default height of 100 mm is clipped.

```
\begin{BackFace}
    ...
\end{BackFace}
```

LeftFace The environment `LeftFace` is used to define the content of the left face. The environment processes its contents inside a `minipage` of the default width 100 mm reduced by the left and right separator width (`faceSep`).

The minipage is centered vertically on the face. Usually anything extending the default height of 30 mm is clipped.

```
\begin{LeftFace}
    ...
\end{LeftFace}
```

RightFace The environment `RightFace` is used to define the content of the right face. The environment processes its contents inside a `minipage` of the default width 100 mm reduced by the left and right separator width (`faceSep`).

The minipage is centered vertically on the face. Usually anything extending the default height of 30 mm is clipped.

```
\begin{RightFace}
    ...
\end{RightFace}
```

TopFace The environment `TopFace` is used to define the content of the top face. The environment processes its contents inside a `minipage` of the default width 88 mm reduced by the left and right separator width (`faceSep`).

The minipage is centered vertically on the face. Usually anything extending the default height of 30 mm is clipped.

```
\begin{TopFace}
    ...
\end{TopFace}
```

BottomFace The environment `BottomFace` is used to define the content of the bottom face.

The environment processes its contents inside a `minipage` of the default width 88 mm reduced by the left and right separator width (`faceSep`).

The minipage is centered vertically on the face. Usually anything extending the default height of 30 mm is clipped.

```
\begin{BottomFace}
    ...
\end{BottomFace}
```

Any of the face defining environments described above can take an optional argument. This argument is used to specify the background. In the simplest case you just have one background color. This is specified with the keyword `color`.

```
\begin{ProductBox}
    \begin{FrontFace}
        [color=yellow]
        ...
    \end{FrontFace}
    ...
\end{ProductBox}
```


Colors in Ti*k*Z are either one of the named colors or a composition of those colors. The notation `red!60!blue` denotes the color by mixing 60% red and 40% blue.

```
\begin{ProductBox}
    \begin{FrontFace}
        [color=red!60!blue]
        ...
    \end{FrontFace}
    ...
\end{ProductBox}
```

By mixing in black or white you can come to a lighter or darker color.

```
\begin{ProductBox}
    \begin{FrontFace}
        [color=red!20!white]
        ...
    \end{FrontFace}
    ...
\end{ProductBox}
```


A fading from top to bottom can be specified with two colors named `top color` and `bottom color`.

```
\begin{ProductBox}
    \begin{FrontFace}
        [top color=red!50!blue,bottom color=yellow]
        ...
    \end{FrontFace}
    ...
\end{ProductBox}
```

A fading from left to right can be specified with two colors named `left color` and `left color`.

```
\begin{ProductBox}
    \begin{FrontFace}
        [left color=red!50!blue,right color=yellow]
        ...
    \end{FrontFace}
    ...
\end{ProductBox}
```

The parameter `middle color` can be used in horizontal or vertical fading to specifying the color in the middle. Note thta it has to be specified after the other colors!

```
\begin{ProductBox}
    \begin{FrontFace}
        [top color=red,bottom color=yellow,middle color=blue]
        ...
    \end{FrontFace}
    ...
\end{ProductBox}
```

A circular fading can be specified with the color names `inner color` and `outer color`.

```
\begin{ProductBox}
    \begin{FrontFace}
        [inner color=red,outer color=yellow]
        ...
    \end{FrontFace}
    ...
\end{ProductBox}
```

2.2 The Extended User Interface

The extended user interface allows you to use all features of Ti*k*Z. For this purpose another set of face defining environments is provided which process their content in a `tikzpicture` environment.

The following example shows a complete example of a product box with the extended user interface. This example is used below to demonstrate the effect of the differnt options.

```
\begin{ProductBox}\sf
  \begin{Front}
    \fill [top color=white!30!blue,bottom color=white]
      rectangle (88mm,100mm);
    \fill [bottom color=white!40!blue,top color=white!90!blue]
          (0mm,0mm) -- (44mm,90mm) -- (88mm,0mm) -- cycle;
    \draw (44mm,40mm) node{\Huge Product Box};
  \end{Front}
  \begin{Back}
    \fill [top color=yellow!30!red,bottom color=white]
      rectangle (88mm,100mm);
    \fill [top color=white,bottom color=blue]
    (0mm,0mm) -- (88mm,100mm) -- (0mm,100mm) --cycle;
    \draw (44mm,50mm) node{\large Back};
  \end{Back}
  \begin{Top}
    \fill [outer color=white!30!red,inner color=white]
      rectangle (88mm,30mm);
    \draw[white,thick] (0mm,5mm) -- (88mm,25mm);
    \draw (44mm,15mm) node{\large Top};
  \end{Top}
  \begin{Bottom}
    \fill [outer color=red!30!white,inner color=black]
      rectangle (88mm,30mm);
    \draw[white] (44mm,15mm) node{\large Bottom};
  \end{Bottom}
  \begin{Left}
    \fill [left color=green,right color=yellow]
      rectangle (30mm,100mm);
    \draw (15mm,50mm) node{\large Left};
  \end{Left}
  \begin{Right}
    \fill [top color=green,bottom color=yellow]
      rectangle (30mm,100mm);
    \draw (15mm,50mm) node{\large Right};
  \end{Right}
\end{ProductBox}
```

The content of the environment is simply expanded. It may contain any code you like – except an `ProductBox` environment. Useful for the production of a product box are some inner environments. They are called `Front`, `Back`, `Left`, `Right`, `Top`, and `Bottom`. They are defined inside the main environment only. Those environments can be used to specify the contents of the respective faces of the box.

Note that in the three-dimensional rendering will show at most three of the faces. It does not hurt to define all of them, even if they are not shown at all.

Front

The environment `Front` is used to define the content of the font face. The environment processes its contents inside a `tikzpicture` of the default size 88 mm×100 mm. Usually anything outside of this range is clipped.

```
\begin{Front}
    ...
\end{Front}
```

Back

The environment `Back` is used to define the content of the back face. The

environment processes its contents inside a `tikzpicture` of the default size 88 mm×100 mm. Usually anything outside of this range is clipped.

```
\begin{Back}
    ...
\end{Back}
```

Left The environment `Left` is used to define the content of the left face, i.e. the face left to the front page. The environment processes its contents inside a `tikzpicture` of the default size 30 mm×100 mm. Usually anything outside of this range is clipped.

```
\begin{Left}
    ...
\end{Left}
```

Right The environment `Right` is used to define the content of the right face, i.e. the face right to the front page. The environment processes its contents inside a `tikzpicture` of the default size 30 mm×100 mm. Usually anything outside of this range is clipped.

```
\begin{Right}
    ...
\end{Right}
```

Top The environment `Top` is used to define the content of the top face. The environment processes its contents inside a `tikzpicture` of the default size 88 mm×30 mm. Usually anything outside of this range is clipped.

```
\begin{Top}
    ...
\end{Top}
```

Bottom The environment `Bottom` is used to define the content of the top face. The environment processes its contents inside a `tikzpicture` of the default size 88 mm×30 mm. Usually anything outside of this range is clipped.

```
\begin{Bottom}
    ...
\end{Bottom}
```

2.3 Settings and Options of the Main Environment

The environment `ProductBox` can take some options to influence the appearance of the product box. Those options are comma separated.

```
\begin{ProductBox}[shape=3d]
```

The settings are local to the main environment. If an option is not set then the fallback from the global settings are used.

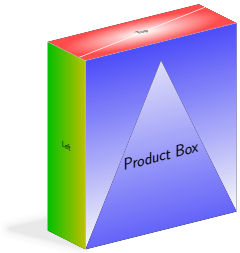
\ProductBoxSet The macro `\ProductBoxSet` modifies the global setting of the product box style. The arguments are the same as the optional arguments of the environment `ProductBox` – but enclosed in braces instead of brackets.

```
\ProductBoxSet{shape=3d}
```

The following options can be used to influence the result of the product box.

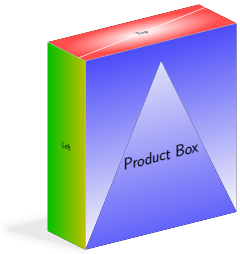
style The box style determines, how the box is drawn. Several rendering functions are provided to produce different effects. Any value is accepted. Unknown box styles will lead to an error message.

```
\begin{ProductBox}[style=3D]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



3D The box style `3D` is the default. It produces a three-dimensional view of the box. The option `3D` is an abbreviation for `style=3D`. It can also be written as `3d` or `threeD`.

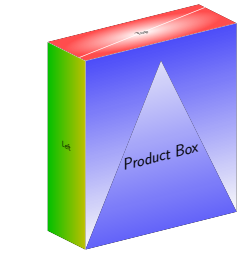
```
\begin{ProductBox}
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



shadow The option `shadow` controls the drawing of the drop shadow in the 3D rendering. It is a boolean value taking the values `true` and `false`. The default value is `true`.

The option `shadow` is the abbreviation for `shadow=true`.

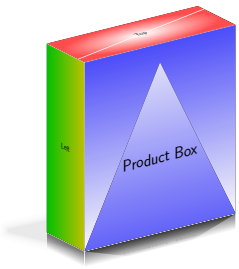
```
\begin{ProductBox}[shadow=false]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



mirror The option `mirror` controls the rendering of the mirror effect in the 3D rendering. It is a boolean value taking the values `true` and `false`. The default value is `false`.

The option `mirror` is the abbreviation for `mirror=true`.

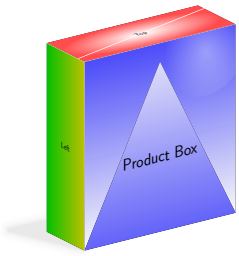
```
\begin{ProductBox}[mirror=true]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



flare The option `flare` controls the rendering of the flare effect in the 3D rendering. The flare is a circular relection of the light source in the upper right corner of the front face. The option is a boolean value taking the values `true` and `false`. The default value is `false`.

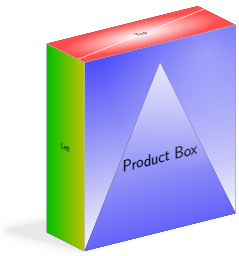
The option `flare` is the abbreviation for `flare=true`.

```
\begin{ProductBox}[flare=true]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



flareDiameter The option `flareDiameter` takes a dimension which defines the diameter of the flare effect in the 3D rendering. The default value is 24 mm.

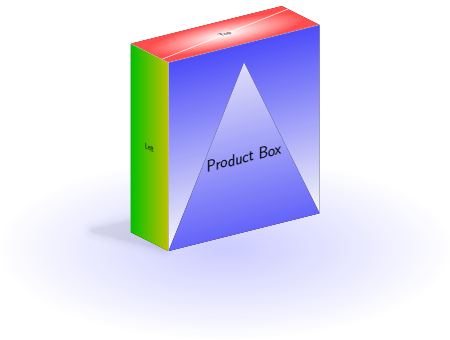
```
\begin{ProductBox}[flare=true,
flareDiameter=60mm]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



Internally the 3D rendering uses a `tikzpicture`. You can expand your own code either at the beginning or at the end of this environment. This can be achieved by overwriting a macro.

ProductBoxThreeDStartHook The macro `ProductBoxThreeDStartHook` contains code to be expanded at the beginning of the 3D rendering. Initially it is defined as empty.

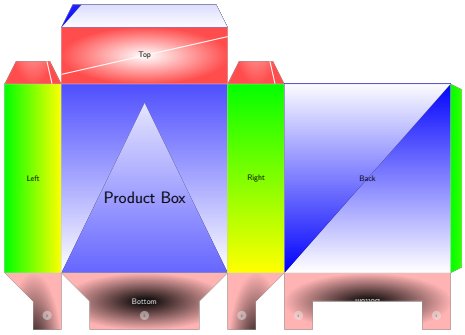
```
\renewcommand\ProductBoxThreeDStartHook{
    \filldraw[yscale=.4,
              shift={(30mm,4mm)},
              color=white,
              inner color=white!70!blue,
              outer color=white]
         circle (120mm);
}
\begin{ProductBox}
    ...
\end{ProductBox}
```



ProductBoxThreeDEndHook The macro `ProductBoxThreeDEndHook` contains code to be expanded at the end of the 3D rendering. Initially it is defined as empty.

fold The option `fold` switches to the box style for rendering a complete wire frame with all faces in place. The option `fold` is an abbreviation for `style=fold`.

```
\begin{ProductBox}[fold]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```

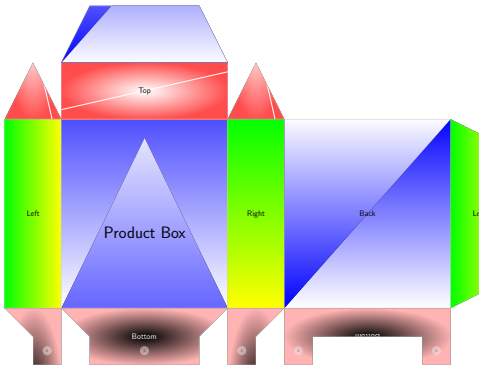


The top face is printed on the left and right top ear as well. This should avoid a break in the pattern when the box is partially opened. The same principle applis for the glue ear on the right side and the bottom.

The bottom is formed in a way to maiximize stability without the need to glue. In addition numbers are printed on the parts of the bottom indicating the sequence in which the parts should be closed.

earSize The option `earSize` takes a dimension which defines the width of the ears in the fold rendering. The ears around the top are this wide. The width of the glueing ear is half of this size. The default value is 12 mm.

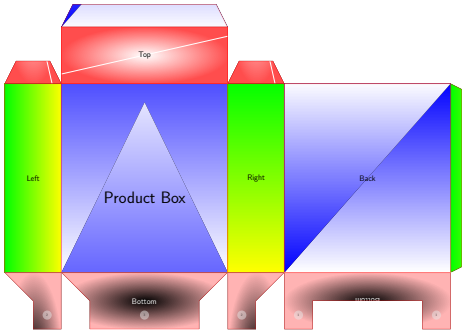
```
\begin{ProductBox}[fold,earSize=30mm]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



Note that the ear size must not exceed the width of the left face, the width of the front face, and the heiht of the box. Otherwise funny effects in the ears will happen.

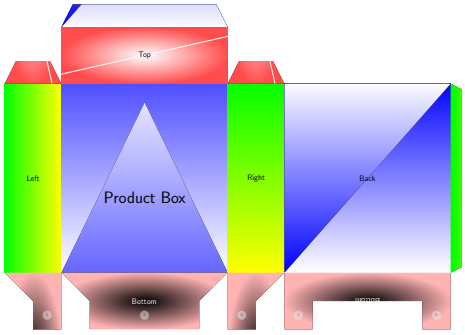
foldLine The option `foldLine` takes a specification for the line surrounding the fold drawing. Usally you want to simply use a color like “gray” or “red”. The default is a kind of gray.

```
\begin{ProductBox}[fold,foldLine=red]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



foldOpacity The option `foldOpacity` takes a fraction for the opacity of the line surrounding the fold drawing. The default is `0.5`. If you want to let the fold lines disappear then use a value of 1.

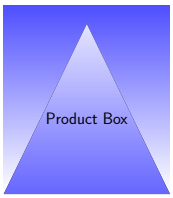
```
\begin{ProductBox}[fold,foldOpacity=0]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



front The option `front` switches to the box style for rendering the front face only. The

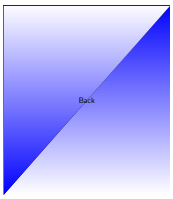
option `front` is an abbreviation for `style=front`.

```
\begin{ProductBox}[front]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



back The option `back` switches to the box style for rendering the back face only. The option `back` is an abbreviation for `style=back`.

```
\begin{ProductBox}[back]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



left The option `left` switches to the box style for rendering the left face only. The option `left` is an abbreviation for `style=left`.

```
\begin{ProductBox}[left]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```

right The option `right` switches to the box style for rendering the right face only. The option `right` is an abbreviation for `style=right`.

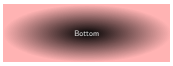
```
\begin{ProductBox}[right]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```

top The option `top` switches to the box style for rendering the top face only. The option `top` is an abbreviation for `style=top`.

```
\begin{ProductBox}[top]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```

bottom The option `bottom` switches to the box style for rendering the face only. The option `bottom` is an abbreviation for `style=bottom`.

```
\begin{ProductBox}[bottom]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



empty The option `empty` switches to the box style for not rendering the box at all. The option `empty` is an abbreviation for `style=empty`.

```
\begin{ProductBox}[empty]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```

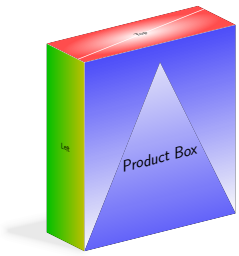
scale The option `scale` controlls the scaling of the whole rendering. It is a number where 1. represents the original size.[2]

```
\begin{ProductBox}[scale=.1]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```

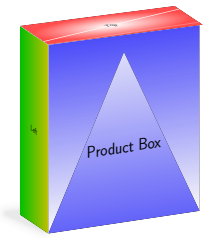
view The option `view` takes a name of a view definition and activates the appropriate settings. A few views are predefined. The default value is `1`.

[2] The examples on the right side are normally rendered with a scale of 0.25.

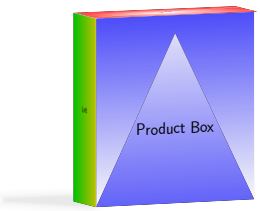
```
\begin{ProductBox}[view=1]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```




```
\begin{ProductBox}[view=2]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```




```
\begin{ProductBox}[view=3]
    \begin{Front}
        ...
    \end{Front}
    \begin{Left}
        ...
    \end{Left}
    ...
\end{ProductBox}
```



edgeColor The option `edgeColor` takes a color specification for highlighting the inner edges in the 3D rendering.

faceSep The option `faceSep` takes a dimension denoting the additional separating whitespace between the outer border and the minipage in the simplified interface.

width The option `width` takes a dimension denoting the width of the box. The default value is 88 mm.

height The option `height` takes a dimension denoting the height of the box. The default value is 100 mm.

depth The option `depth` takes a dimension denoting the depth of the box. The default value is 30 mm.

clean The option `clean` controlls the cleaning of the stored faces upon entering the main environment. If it is turned off then the previously defined faces are still present and do not need to be defined again. It is a boolean value taking the values `true` and `false`. The default value is `true`.

This option can be used to typeset the same product box with differnt parameters. For this purpose the main environment `ProductBox` is left empty and the option `clean=false` is added.

clip The option `clip` controls the clipping of the faces to their defined size. If it is

turned off then the faces can be oversized leading to strange effects. It is a boolean value taking the values `true` and `false`. The default value is `true`.

3 Tipps and Tricks

3.1 Adjusting the Paper for the Fold Rendering

When you produce the fold rendering it is usually meant to be cut out and glued together. This means that the normal rules for the paper layout are not relevant. Instead you want to use the complete page for printing the product box. Below an example is shown how this goal can be achieved.

```
\documentclass{report}
\usepackage[a4paper,
            landscape,
            left=5mm,
            right=5mm,
            top=5mm,
            bottom=5mm]{geometry}
\usepackage{productbox}
% load any required packages here
\pagestyle{empty}
\begin{document}
\noindent
\begin{ProductBox}[style=fold]
  % include any definitions for the faces here
\end{ProductBox}
\end{document}
```

The class for typesetting this example is `report`. This can be changed to suit your needs. For instance if you are used to a document class with other macros and environments predefined you can just use it instead.

The example above uses the package `geometry` [Ume07] to get rid of any predefined page layout. Some parameters (marked in red) can be adjusted. First of all is the paper definition. Here the value `a4paper` is used. If you want to print onto paper of a different size just use an appropriate short name like `letterpaper` or `a3paper`. See the documentation of the geometry package for a complete list of values.

The values `left`, `right`, `top`, and `bottom` denote the margins left on the respective outer side of the paper. They are set to 5 mm in this example to cope with the problem that some printers are not able to fill the complete page. They might need a small non-printable area at the borders. You can experiment and adjust those values to whatever suits your printer.

4 Known Problems

This section lists some issues which might lead to undesirable results.

Nested `ProductBox` environments. The definition of the environment uses some global storage. As a consequence the environment `ProductBox` can not be used inside the definition of a face. For instance if you want to show a product box on a side of another product box can lead to this problem.

In this case you can simply store the inner product box in a box register (with `setbox`) and use this bos register instead of a direct rendering. This will overcome the restriction.

Free selection of the point of view. The 3D rendering is rather limited in the possibilities of selecting the view. Arbitrary rotation about any axis is not implemented (yet). This is on the which list for a future release already.

References

[Tan08] Till Tantau. *TikZ & PGF: Manual for Version 2.0*, February 2008. `http://sourceforge.net/projects/pgf`.

[Ume07] Hideo Umeki. *The* `geometry` *package*, July 2007.

5 The Documentation Driver

The documentation driver changes productbox.dtx into a self-extracting documentation. Thus it is possible to run LaTeX on productbox.dtx to produce the documentation.

The documentation can be adapted in a file named productbox.dcf (documentation configuration). This file can contain instructions for `docstrip`. Especially useful might be the instruction

`\OnlyDescription`

which suppresses the generation of the implementation description.

```
⟨*driver⟩
\documentclass{ltxdoc}
\usepackage{productbox}
\usepackage[colorlinks,citecolor=blue]{hyperref}
\usepackage{graphicx,color}
\RecordChanges
\EnableCrossrefs
\CodelineIndex
\definecolor{darkblue}{rgb}{.4,.4,1.}
\renewcommand\MacroFont{\tt\footnotesize\color{darkblue}}
\parindent=0pt
\parskip=1ex plus .5ex minus .25ex
\InputIfFileExists{productbox.dcf}{}{}
\begin{document}
\DocInput{productbox.dtx}
\end{document}
⟨/driver⟩
```

6 The Implementation

The implementation contains the code of the style.

6.1 The Version Information

The following lines define the version information for the class file. The information is partially taken from the version control system (Subversion).

```
⟨*style⟩
\begingroup
  \def\ProductBox@VC$#1: #2 #3${#2}
  \def\ProductBox@VCdate$#1: #2-#3-#4 #5${#2/#3/#4}
  \xdef\fileversion{1.1}
  \xdef\filerevision{\ProductBox@VC$Revision: 8333 $}
  \xdef\filedate{\ProductBox@VCdate
    $Date: 2010-12-29 20:58:06 +0100 (Mi, 29 Dez 2010) $}
  \xdef\filename{productbox.dtx}
\endgroup
⟨/style⟩
```

6.2 Getting Started

First we have to determine that the right kind of LaTeX is running and identify the style file.

```
⟨*style⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{productbox}[\filedate Another type of boxes...]
```

A bunch of packages is loaded to form the base of the work herein.

```
\RequirePackage{keyval}
\RequirePackage{tikz}
\usetikzlibrary{calc}
\usetikzlibrary{fadings}
```

6.3 Option Declarations

Define the parameters for the keyval package. They are used in the main environemnt `Product Box` and in the declaration of global options `\ProductBoxSet`.

6.3.1 General Parameters

```
\define@key{ProductBox}{scale}{%
  \def\ProductBox@scale{#1}}
\define@key{ProductBox}{width}{%
  \def\ProductBox@x{#1}}
\define@key{ProductBox}{height}{%
  \def\ProductBox@y{#1}}
\define@key{ProductBox}{depth}{%
  \def\ProductBox@z{#1}}
\define@key{ProductBox}{clean}[true]{%
  \csname ProductBox@clean#1\endcsname}
```

6.3.2 Box Style Parameters

```
\define@key{ProductBox}{flat}[true]{%
  \def\ProductBox@style{flat}}
\define@key{ProductBox}{fold}[true]{%
  \def\ProductBox@style{fold}}
\define@key{ProductBox}{3d}[true]{%
  \def\ProductBox@style{threeD}}
\define@key{ProductBox}{3D}[true]{%
  \def\ProductBox@style{threeD}}
\define@key{ProductBox}{threeD}[true]{%
  \def\ProductBox@style{threeD}}
\define@key{ProductBox}{top}[true]{%
  \def\ProductBox@style{top}}
\define@key{ProductBox}{bottom}[true]{%
  \def\ProductBox@style{bottom}}
\define@key{ProductBox}{front}[true]{%
  \def\ProductBox@style{front}}
\define@key{ProductBox}{back}[true]{%
  \def\ProductBox@style{back}}
```

```
\define@key{ProductBox}{left}[true]{%
  \def\ProductBox@style{left}}
\define@key{ProductBox}{right}[true]{%
  \def\ProductBox@style{right}}
\define@key{ProductBox}{empty}[true]{%
  \def\ProductBox@style{empty}}
\define@key{ProductBox}{style}{%
  \def\ProductBox@style{#1}}
```

6.3.3 Parameters for the 3D Rendering

```
\define@key{ProductBox}{shadow}[true]{%
  \csname ProductBox@shadow#1\endcsname}
\define@key{ProductBox}{mirror}[true]{%
  \csname ProductBox@mirror#1\endcsname}
\define@key{ProductBox}{flare}[true]{%
  \csname ProductBox@flare#1\endcsname}
\define@key{ProductBox}{flareDiameter}{%
  \def\ProductBox@flareDiameter{#1}}
\define@key{ProductBox}{edgeColor}{%
  \def\productBox@edgeColor{#1}}
\define@key{ProductBox}{view}{%
  \@nameuse{ProductBox@View@#1}}
```

6.3.4 Parameters for the Fold Rendering

```
\define@key{ProductBox}{earSize}{%
  \def\ProductBox@earSize{#1}}
\define@key{ProductBox}{foldLine}{%
  \def\ProductBox@foldLine{#1}}
\define@key{ProductBox}{foldOpacity}{%
  \def\ProductBox@foldOpacity{#1}}
```

6.3.5 Parameters for the Simplified Interface

```
\define@key{ProductBox}{faceSep}{%
  \def\ProductBox@FaceSep{#1}}
```

6.4 Storage for the Faces

\ProductBox@Front

The box `\ProductBox@Front` contains the front material.

```
\newbox\ProductBox@Front
```

\ProductBox@Left

The box `\ProductBox@Left` contains the left material.

```
\newbox\ProductBox@Left
```

\ProductBox@Top

The box `\ProductBox@Top` contains the top material.

```
\newbox\ProductBox@Top
```

\ProductBox@Right

The box `\ProductBox@Right` contains the right material.

```
\newbox\ProductBox@Right
```

\ProductBox@Bottom The box \ProductBox@Bottom contains the bottom material.

```
96 \newbox\ProductBox@Bottom
```

\ProductBox@Back The box \ProductBox@Back contains the back material.

```
97 \newbox\ProductBox@Back
```

6.5 Settings

\ProductBoxSet The macro \ProductBoxSet defines the global parameters used by the environment ProductBox. They can be overwritten either within a group or in the optional parameter of the environment.

```
98 \newcommand\ProductBoxSet[1]{\setkeys{ProductBox}{#1}}
```

6.6 The Main Environment

\ProductBox@style The default style is the 3D rendering.

```
99 \newcommand\ProductBox@style{threeD}
```

\ProductBox@x The width of the box.

```
100 \newcommand\ProductBox@x{88mm}
```

\ProductBox@y The height of the box.

```
101 \newcommand\ProductBox@y{100mm}
```

\ProductBox@z The depth of the box.

```
102 \newcommand\ProductBox@z{30mm}
```

ifProductBox@clean Indicator that the boxes for the faces should be cleaned.

```
103 \newif\ifProductBox@clean
```

ifProductBox@active Indicator that we are inside a ProductBox environment already.

```
104 \newif\ifProductBox@active
```

ifProductBox@clip The boolean ProductBox@clip determines whether or not the additional clipping of the boxes should be enabled.

```
105 \newif\ifProductBox@clip \ProductBox@cliptrue
```

\ProductBox@FaceSep The horizontal separator of the minipage in the face definitions in the simplified interface.

```
\newcommand\ProductBox@FaceSep{1em}
```

ProductBox This is the central environment provided by this style. In the begin code only the local environments are initialized and the optional parameters are evaluated with the help of the package `keyval`.

```
\newenvironment{ProductBox}[1][]{%
  \setkeys{ProductBox}{#1}%
```

Next we clean the faces if this is required.

```
  \ifProductBox@clean
    \global\setbox\ProductBox@Front\hbox{}%
    \global\setbox\ProductBox@Back\hbox{}%
    \global\setbox\ProductBox@Left\hbox{}%
    \global\setbox\ProductBox@Right\hbox{}%
    \global\setbox\ProductBox@Top\hbox{}%
    \global\setbox\ProductBox@Bottom\hbox{}%
  \fi
```

Next we define the local environments to make sure that they have the proper definitions within this environment. Since the environment provides an implicit group, the definitions are local to this environment.

To suppress any error messages about environments which are already defined the start macros are reset to undefined.

```
  \ifProductBox@active
    \errmessage{Trying to use an environment ProductBox inside the
      environment ProductBox. This is not allowed.}%
  \else
    \ProductBox@activetrue
  \fi
  \let\Front\undefined            \let\endFront\undefined
  \let\Back\undefined             \let\endBack\undefined
  \let\Left\undefined             \let\endLeft\undefined
  \let\Right\undefined            \let\endRight\undefined
  \let\Top\undefined              \let\endTop\undefined
  \let\Bottom\undefined           \let\endBottom\undefined
  \let\FrontFace\undefined        \let\endFrontFace\undefined
  \let\BackFace\undefined         \let\endBackFace\undefined
  \let\LeftFace\undefined         \let\endLeftFace\undefined
  \let\RightFace\undefined        \let\endRightFace\undefined
  \let\TopFace\undefined          \let\endTopFace\undefined
  \let\BottomFace\undefined       \let\endBottomFace\undefined
  \newenvironment{Front}{\ProductBox@Start\ProductBox@Front
    (\ProductBox@x,\ProductBox@y)}{\ProductBox@End}%
  \newenvironment{Back}{\ProductBox@Start\ProductBox@Back
    (\ProductBox@x,\ProductBox@y)}{\ProductBox@End}%
  \newenvironment{Left}{\ProductBox@Start\ProductBox@Left
    (\ProductBox@z,\ProductBox@y)}{\ProductBox@End}%
  \newenvironment{Right}{\ProductBox@Start\ProductBox@Right
    (\ProductBox@z,\ProductBox@y)}{\ProductBox@End}%
```

```
  \newenvironment{Top}{\ProductBox@Start\ProductBox@Top
    (\ProductBox@x,\ProductBox@z)}{\ProductBox@End}%
  \newenvironment{Bottom}{\ProductBox@Start\ProductBox@Bottom(
    \ProductBox@x,\ProductBox@z)}{\ProductBox@End}%
  \newenvironment{FrontFace}[1][white]{\ProductBox@StartFace\ProductBox@Front
    {\ProductBox@x}{\ProductBox@y}{##1}}{\ProductBox@EndFace}%
  \newenvironment{BackFace}[1][white]{\ProductBox@StartFace\ProductBox@Back
    {\ProductBox@x}{\ProductBox@y}{##1}}{\ProductBox@EndFace}%
  \newenvironment{LeftFace}[1][white]{\ProductBox@StartFace\ProductBox@Left
    {\ProductBox@z}{\ProductBox@y}{##1}}{\ProductBox@EndFace}%
  \newenvironment{RightFace}[1][white]{\ProductBox@StartFace\ProductBox@Right
    {\ProductBox@z}{\ProductBox@y}{##1}}{\ProductBox@EndFace}%
  \newenvironment{TopFace}[1][white]{\ProductBox@StartFace\ProductBox@Top
    {\ProductBox@x}{\ProductBox@z}{##1}}{\ProductBox@EndFace}%
  \newenvironment{BottomFace}[1][white]{\ProductBox@StartFace\ProductBox@Bottom
    {\ProductBox@x}{\ProductBox@z}{##1}}{\ProductBox@EndFace}%
  \ignorespacesafterend
}{%
```

Check that the box style is defined or issue an appropriate error message.

```
  \@ifundefined{ProductBox@style@\ProductBox@style}{%
    \errmessage{Box style '\ProductBox@style' for ProductBox is unknown}%
  }{%
```

The main activity is performed in the end code. Since the flexibility of the environment is one of its design goals, the expansion of the macro `\ProductBox@style` is used to invoke the macro stored in it. Optionally it is enclosed in a `\scalebox` macro to perform the scaling. Thus the implementations of the box styles do not need to care about scaling at all.

```
    \@ifundefined{ProductBox@scale}{%
      \@nameuse{ProductBox@style@\ProductBox@style}}{%
      \scalebox{\ProductBox@scale}%
               {\@nameuse{ProductBox@style@\ProductBox@style}}
    }}%
  \ProductBox@activefalse
  \ignorespacesafterend
}
```

`\ProductBox@Start` The macro `\ProductBox@Start` starts the environment storing a face.

```
\def\ProductBox@Start#1(#2){%
  \global\setbox#1\hbox\bgroup\begin{tikzpicture}%
    \ifProductBox@clip \clip rectangle (#2); \fi
  }%
```

`\ProductBox@End` The macro `\ProductBox@End` ends the environment storing a face.

```
\newcommand\ProductBox@End{\end{tikzpicture}\egroup\ignorespacesafterend}
```

`\ProductBox@StartFace` The macro `\ProductBox@StartFace` starts the environment storing a face in a minipage.

```
\newcommand\ProductBox@StartFace[4]{%
```

```
  \let\ProductBox@@box#1%
  \def\ProductBox@@w{#2}%
  \def\ProductBox@@h{#3}%
  \def\ProductBox@@args{[#4]}%
  \@tempdima=#2
  \advance\@tempdima-\ProductBox@FaceSep
  \advance\@tempdima-\ProductBox@FaceSep
  \setbox#1\hbox\bgroup\begin{minipage}{\the\@tempdima}%
    \ignorespaces
}
```

`\ProductBox@EndFace` This macro ends a face definition and stores the result in the internal box `\ProductBox@@box`.

```
\newcommand\ProductBox@EndFace{\end{minipage}\egroup
    \global\setbox\ProductBox@@box=\hbox{\begin{tikzpicture}
    \ifProductBox@clip \clip rectangle (\ProductBox@@w,\ProductBox@@h); \fi
    \expandafter\fill \ProductBox@@args rectangle(\ProductBox@@w,\ProductBox@@h);
    \draw (.5*\ProductBox@@w,.5*\ProductBox@@h)
      node[inner sep=0pt] {\box\ProductBox@@box};
    \end{tikzpicture}}\ignorespacesafterend}
```

6.7 Box Styles

`\ProductBox@style@empty` This macro defines the sytle *empty* for a product box. It simply does nothing.

```
\newcommand\ProductBox@style@empty{}
```

`\ProductBox@style@flat` This macro defines the sytle *flat* for a product box.

```
\newcommand\ProductBox@style@flat{%
  \par\noindent
  \hspace*{\ProductBox@z}\kern1mm \copy\ProductBox@Top\\[1ex]
  \copy\ProductBox@Left\kern1mm \copy\ProductBox@Front\kern1mm
  \copy\ProductBox@Right\kern1mm \copy\ProductBox@Back\\[1ex]
  \hspace*{\ProductBox@z}\kern1mm \copy\ProductBox@Bottom\par
}
```

6.7.1 Box Style `front`

`\ProductBox@style@front` This macro defines the sytle *front* for a product box. Only this one face of the box is shown.

```
\newcommand\ProductBox@style@front{%
  \copy\ProductBox@Front
}
```

6.7.2 Box Style `back`

`\ProductBox@style@back` This macro defines the sytle *back* for a product box. Only this one face of the box is shown.

```
\newcommand\ProductBox@style@back{%class
  \copy\ProductBox@Back
}
```

6.7.3 Box Style `left`

`\ProductBox@style@left` This macro defines the sytle *left* for a product box. Only this one face of the box is shown.

```
\newcommand\ProductBox@style@left{%
  \copy\ProductBox@Left
}
```

6.7.4 Box Style `right`

`\ProductBox@style@right` This macro defines the sytle *right* for a product box. Only this one face of the box is shown.

```
\newcommand\ProductBox@style@right{%
  \copy\ProductBox@Right
}
```

6.7.5 Box Style `top`

`\ProductBox@style@top` This macro defines the sytle *top* for a product box. Only this one face of the box is shown.

```
\newcommand\ProductBox@style@top{%
  \copy\ProductBox@Top
}
```

6.7.6 Box Style `bottom`

`\ProductBox@style@bottom` This macro defines the sytle *bottom* for a product box. Only this one face of the box is shown.

```
\newcommand\ProductBox@style@bottom{%
  \copy\ProductBox@Bottom
}
```

6.7.7 Box Style `fold`

\ProductBox@earSize Parameter for the size of the ears.

```
\newcommand\ProductBox@earSize{12mm}
```

\ProductBox@foldLine The color of the additional lines in the fold rendering.

```
\newcommand\ProductBox@foldLine{white!80!black}
```

\ProductBox@foldOpacity The color of the additional lines in the fold rendering.

```
\newcommand\ProductBox@foldOpacity{.5}
```

\ProductBox@style@fold This macro defines the sytle *fold* for a product box.

```
\newcommand\ProductBox@style@fold{%
  \begin{tikzpicture}[sw/.style={anchor=south west,
                                 inner sep=0pt},
                      se/.style={anchor=south east,
                                 inner sep=0pt},
                      nw/.style={anchor=north west,
                                 inner sep=0pt},
                      num/.style={circle,
                                  fill=white!90!black,
                                  fill opacity=.5,
                                  font=\tiny\bfseries\sffamily}]
```

left face

```
    \begin{scope}
      \clip rectangle (\ProductBox@z,\ProductBox@y);
      \draw node[sw]{\copy\ProductBox@Left};
      \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
        rectangle (\ProductBox@z,\ProductBox@y);
    \end{scope}
```

front face

```
    \begin{scope}[xshift=\ProductBox@z]
      \clip rectangle (\ProductBox@x,\ProductBox@y);
      \draw node[sw]{\copy\ProductBox@Front};
      \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
        rectangle (\ProductBox@x,\ProductBox@y);
    \end{scope}
```

right face

```
    \begin{scope}[xshift=\ProductBox@z+\ProductBox@x]
      \clip rectangle (\ProductBox@z,\ProductBox@y);
      \draw node[sw]{\copy\ProductBox@Right};
      \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
        rectangle (\ProductBox@z,\ProductBox@y);
    \end{scope}
```

back face

```
\begin{scope}[xshift=\ProductBox@x+2*\ProductBox@z]
  \clip rectangle (\ProductBox@x,\ProductBox@y);
  \draw node[sw]{\copy\ProductBox@Back};
  \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
     rectangle (\ProductBox@x,\ProductBox@y);
\end{scope}
```

top face

```
\begin{scope}[xshift=\ProductBox@z,yshift=\ProductBox@y]
  \clip rectangle (\ProductBox@x,\ProductBox@z);
  \draw node[sw]{\copy\ProductBox@Top};
  \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
    rectangle (\ProductBox@x,\ProductBox@z);
\end{scope}
```

bottom ear 3 front

```
\begin{scope}[xshift=\ProductBox@z]
  \clip
    (0mm,0mm) -- (.5*\ProductBox@z,-.5*\ProductBox@z) --
    (.5*\ProductBox@z,-\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-.5*\ProductBox@z) --
    (\ProductBox@x,0mm) -- cycle;
  \draw (\ProductBox@x,-\ProductBox@z)
    node[se]{\copy\ProductBox@Bottom};
  \draw [thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
    (0mm,0mm) -- (.5*\ProductBox@z,-.5*\ProductBox@z) --
    (.5*\ProductBox@z,-\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-.5*\ProductBox@z) --
    (\ProductBox@x,0mm) -- cycle;
  \draw (.5*\ProductBox@x,-.75*\ProductBox@z) node[num]{3};
\end{scope}
```

bottom ear 1 back

```
\begin{scope}[xshift=\ProductBox@x+2*\ProductBox@z]
  \clip
    (0mm,0mm) -- (0mm,-\ProductBox@z) --
    (.5*\ProductBox@z,-\ProductBox@z) --
    (.5*\ProductBox@z,-.5*\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-.5*\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-\ProductBox@z) --
    (\ProductBox@x,-\ProductBox@z) -- (\ProductBox@x,0mm) --
    cycle;
  \draw node[se,rotate=180]{\copy\ProductBox@Bottom};
  \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
    (0mm,0mm) -- (0mm,-\ProductBox@z) --
    (.5*\ProductBox@z,-\ProductBox@z) --
    (.5*\ProductBox@z,-.5*\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-.5*\ProductBox@z) --
    (\ProductBox@x-.5*\ProductBox@z,-\ProductBox@z) --
    (\ProductBox@x,-\ProductBox@z) -- (\ProductBox@x,0mm) --
    cycle;
  \draw (.25*\ProductBox@z,-.75*\ProductBox@z) node[num]{1};
```

```
    \draw (\ProductBox@x-.25*\ProductBox@z,-.75*\ProductBox@z)
      node[num]{1};
  \end{scope}
```

bottom ear 2 right

```
  \begin{scope}[xshift=\ProductBox@z+\ProductBox@x]
    \clip
      (0mm,0mm) -- (\ProductBox@z,0mm) --
      (.5*\ProductBox@z,-.5*\ProductBox@z) --
      (.5*\ProductBox@z,-\ProductBox@z) -- (0mm,-\ProductBox@z) --
      cycle;
    \draw (\ProductBox@z,0mm)
      node[se,rotate=90]{\copy\ProductBox@Bottom};
    \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
      (0mm,0mm) -- (\ProductBox@z,0mm) --
      (.5*\ProductBox@z,-.5*\ProductBox@z) --
      (.5*\ProductBox@z,-\ProductBox@z) -- (0mm,-\ProductBox@z) --
      cycle;
    \draw (.25*\ProductBox@z,-.75*\ProductBox@z) node[num]{2};
  \end{scope}
```

bottom ear 2 left

```
  \begin{scope}
    \clip
      (0mm,0mm) -- (\ProductBox@z,0mm) --
      (\ProductBox@z,-\ProductBox@z) --
      (.5*\ProductBox@z,-\ProductBox@z) --
      (.5*\ProductBox@z,-.5*\ProductBox@z) -- cycle;
    \draw
      node[sw,rotate=270]{\copy\ProductBox@Bottom};
    \draw [thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
      (0mm,0mm) -- (\ProductBox@z,0mm) --
      (\ProductBox@z,-\ProductBox@z) --
      (.5*\ProductBox@z,-\ProductBox@z) --
      (.5*\ProductBox@z,-.5*\ProductBox@z) -- cycle;
    \draw (.75*\ProductBox@z,-.75*\ProductBox@z) node[num]{2};
  \end{scope}
```

glue ear

```
  \begin{scope}[xshift=2*\ProductBox@x+2*\ProductBox@z]
    \clip
      (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize/4) --
      (\ProductBox@earSize/2,\ProductBox@y-\ProductBox@earSize/4) --
      (0mm,\ProductBox@y) -- cycle;
    \draw node[sw]{\copy\ProductBox@Left};
    \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
      (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize/4) --
      (\ProductBox@earSize/2,\ProductBox@y-\ProductBox@earSize/4) --
      (0mm,\ProductBox@y) -- cycle;
  \end{scope}
```

top ear left

```
  \begin{scope}[yshift=\ProductBox@y]
    \clip
      (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize) --
```

```
        (\ProductBox@z-\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@z,0mm) -- cycle;
      \draw node[nw,rotate=90]{\copy\ProductBox@Top};
      \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
        (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@z-\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@z,0mm) -- cycle;
    \end{scope}
```

top ear right

```
    \begin{scope}[xshift=\ProductBox@x+\ProductBox@z,
                  yshift=\ProductBox@y]
      \clip
        (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@z-\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@z,0mm) -- cycle;
      \draw node[se,rotate=270]{\copy\ProductBox@Top};
      \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
        (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@z-\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@z,0mm) -- cycle;
    \end{scope}
```

top ear back

```
    \begin{scope}[xshift=\ProductBox@z,
                  yshift=\ProductBox@y+\ProductBox@z]
      \clip
        (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@x-\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@x,0mm) -- cycle;
      \draw (0mm,\ProductBox@y)
        node[se,rotate=180]{\copy\ProductBox@Back};
      \draw[thin,opacity=\ProductBox@foldOpacity,\ProductBox@foldLine]
        (0mm,0mm) -- (\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@x-\ProductBox@earSize/2,\ProductBox@earSize) --
        (\ProductBox@x,0mm) -- cycle;
    \end{scope}
  \end{tikzpicture}}
```

6.7.8 Box Style `threeD`

`\ProductBoxThreeDStartHook` A hook to add some code at the beginning.

```
\newcommand\ProductBoxThreeDStartHook{}
```

`\ProductBoxThreeDEndHook` A hook to add some code at the end.

```
\newcommand\ProductBoxThreeDEndHook{}
```

`\ifProductBox@shadow` The boolean `ProductBox@shadow` determines whether or not the shadow in the 3D rendering is shown.

```
\newif\ifProductBox@shadow \ProductBox@shadowtrue
```

\ifProductBox@mirror The boolean `ProductBox@mirror` determines whether or not the mirror effect in the 3D rendering is shown.

```
\newif\ifProductBox@mirror
```

\ifProductBox@flare The boolean `ProductBox@flare` determines whether or not the flare effect in the 3D rendering is shown.

```
\newif\ifProductBox@flare
```

\ProductBox@edgeColor The edge color in the 3D rendering.

```
\newcommand\ProductBox@edgeColor{white}
```

\ProductBox@flareDiameter The diameter of the flare.

```
\newcommand\ProductBox@flareDiameter{24mm}
```

The angle to rotate the box about the z axis.

```
\newcommand\ProductBox@angleZ{8}
```

The angle to rotate the box about the x axis.

```
\newcommand\ProductBox@angleX{10}
```

\ProductBox@style@threeD This macro defines the sytle *threeD* for a product box. The faces are placed such the illusion of a three-dimensional box appears. In addition a shadow is shown if not disabled.

```
\newcommand\ProductBox@style@threeD{
  \begin{tikzpicture}[sw/.style={anchor=south west,
                                 inner sep=0pt}]
    \ProductBoxThreeDStartHook
    \ifProductBox@mirror \ProductBox@threeD@mirror \fi
    \ifProductBox@shadow \ProductBox@threeD@shadow \fi

    \begin{scope}
      \fill[white,
            yslant=\ProductBox@p@front@yslant,
            xscale=\ProductBox@p@front@xscale,
            yscale=\ProductBox@p@front@yscale,
            anchor=south west]
        rectangle (\ProductBox@x,\ProductBox@y);
      \draw
        node[sw,
             yslant=\ProductBox@p@front@yslant,
             xscale=\ProductBox@p@front@xscale,
             yscale=\ProductBox@p@front@yscale]
            {\copy\ProductBox@Front};
      \fill[black,
            fill opacity=.025,
            yslant=\ProductBox@p@front@yslant,
            xscale=\ProductBox@p@front@xscale,
```

```
        yscale=\ProductBox@p@front@yscale,
        anchor=south west]
      rectangle (\ProductBox@x,\ProductBox@y);
    \ifProductBox@flare
      \begin{scope}[
        yslant=\ProductBox@p@front@yslant,
        xscale=\ProductBox@p@front@xscale,
        yscale=\ProductBox@p@front@yscale]
       \clip rectangle(\ProductBox@x,\ProductBox@y);
       \ProductBox@threeD@flare
      \end{scope}
    \fi
  \end{scope}
  \begin{scope}[shift={(-\ProductBox@p@left@xscale*\ProductBox@z,
                       \ProductBox@p@left@yslant
                          *\ProductBox@p@left@xscale
                          *\ProductBox@z)}]
    \fill[white,
        yslant=-\ProductBox@p@left@yslant,
        xscale=\ProductBox@p@left@xscale,
        yscale=\ProductBox@p@left@yscale,
        anchor=south west]
      rectangle (\ProductBox@z,\ProductBox@y);
    \draw
      node[sw,
         yslant=-\ProductBox@p@left@yslant,
         xscale=\ProductBox@p@left@xscale,
         yscale=\ProductBox@p@left@yscale]
        {\copy\ProductBox@Left};
    \fill[black,
        fill opacity=.25,
        yslant=-\ProductBox@p@left@yslant,
        xscale=\ProductBox@p@left@xscale,
        yscale=\ProductBox@p@left@yscale,
        anchor=south west]
      rectangle (\ProductBox@z,\ProductBox@y);
  \end{scope}
  \begin{scope}[yshift=\ProductBox@y]
    \fill [white,
         anchor=south west,
         xslant=-\ProductBox@p@top@xslant,
         xscale=\ProductBox@p@top@xscale,
         yslant=\ProductBox@p@top@yslant,
         yscale=\ProductBox@p@top@yscale]
      rectangle (\ProductBox@x,\ProductBox@z);
    \draw
      node[sw,
         xslant=-\ProductBox@p@top@xslant,
         xscale=\ProductBox@p@top@xscale,
         yslant=\ProductBox@p@top@yslant,
         yscale=\ProductBox@p@top@yscale]
        {\copy\ProductBox@Top};
    \fill [black,
         fill opacity=.0025,
         anchor=south west,
         xslant=-\ProductBox@p@top@xslant,
         xscale=\ProductBox@p@top@xscale,
```

```
            yslant=\ProductBox@p@top@yslant,
            yscale=\ProductBox@p@top@yscale]
         rectangle (\ProductBox@x,\ProductBox@z);
    \end{scope}
    \begin{scope}[shift={(-\ProductBox@p@left@xscale*\ProductBox@z,
        \ProductBox@p@left@yslant*\ProductBox@p@left@xscale*\ProductBox@z)}]
      \draw [\ProductBox@edgeColor,
            line width=.5mm,
            draw opacity=.25,
            yslant=-\ProductBox@p@left@yslant,
            xscale=\ProductBox@p@left@xscale]
        (\ProductBox@z,0mm) --
        (\ProductBox@z,\ProductBox@y) --
        (0mm,\ProductBox@y);
    \end{scope}
    \draw [\ProductBox@edgeColor,
           line width=.5mm,
           draw opacity=.25,
           yslant=\ProductBox@p@front@yslant,
           xscale=\ProductBox@p@front@xscale,
           anchor=south west]
      (0mm,\ProductBox@y) -- (\ProductBox@x,\ProductBox@y);
    \ProductBoxThreeDEndHook
  \end{tikzpicture}}
```

Parameters and views.

```
\newcommand\ProductBox@setFrontParams[3]{%
  \def\ProductBox@p@front@yslant{#1}%
  \def\ProductBox@p@front@xscale{#2}%
  \def\ProductBox@p@front@yscale{#3}%
}
\newcommand\ProductBox@setLeftParams[3]{%
  \def\ProductBox@p@left@yslant{#1}%
  \def\ProductBox@p@left@xscale{#2}%
  \def\ProductBox@p@left@yscale{#3}%
}
\newcommand\ProductBox@setTopParams[4]{%
  \def\ProductBox@p@top@xslant{#1}%
  \def\ProductBox@p@top@yslant{#2}%
  \def\ProductBox@p@top@xscale{#3}%
  \def\ProductBox@p@top@yscale{#4}%
}
\newcommand\ProductBox@setMirrorParams[1]{%
  \def\ProductBox@p@mirror@yscale{#1}%
}
\@namedef{ProductBox@View@1}{%
  \ProductBox@setFrontParams{.25}{.9090909}{1}%
  \ProductBox@setLeftParams{.5}{.6666666}{1}%
  \ProductBox@setTopParams{2}{.2265}{1.363}{.3333333}%
  \ProductBox@setMirrorParams{.4}}
\@namedef{ProductBox@View@2}{%
  \ProductBox@setFrontParams{.15}{.9090909}{1}%
  \ProductBox@setLeftParams{.6666666}{.5}{1}%
  \ProductBox@setTopParams{1.5}{.12}{1.11}{.333333}%
  \ProductBox@setMirrorParams{.4}}
\@namedef{ProductBox@View@3}{%
```

```
  \ProductBox@setFrontParams{.05}{.95}{1}%
  \ProductBox@setLeftParams{.25}{.4}{1}%
  \ProductBox@setTopParams{4}{.04}{1.133333}{.1}%
  \ProductBox@setMirrorParams{.4}}
\@nameuse{ProductBox@View@1}
```

`\ProductBox@threeD@shadow` This macro defines the code to produce the shadow effect for the 3D rendering.

```
\newcommand\ProductBox@threeD@shadow{
    \begin{scope}
      \foreach \x in {.5,1,1.5,2,2.5,3,4,5,6,8,10} {
        \filldraw [black,
                   line width=\x mm,
                   rounded corners=2mm,
                   opacity=.01,
                   shift={(-1mm,1mm)}]
          (0mm,0mm) --
          (-\ProductBox@p@top@xslant
           *\ProductBox@p@left@xscale
           *\ProductBox@z,
           \ProductBox@p@top@yscale
           *\ProductBox@z) --
          (0mm,
           \ProductBox@p@top@yscale*\ProductBox@z +
           \ProductBox@p@top@xslant
           *\ProductBox@p@left@xscale
           *\ProductBox@p@top@yslant
           *\ProductBox@z) --
          cycle;
      }
    \end{scope}
}
```

The following fading is used for the mirror effect in the 3D rendering.

```
\tikzfading[name=ProductBoxFade,
            top color=transparent!100,
            bottom color=transparent!50,
            middle color=transparent!100]
```

`\ProductBox@threeD@mirror` This macro defines the code to produce the mirror effect.

```
\newcommand\ProductBox@threeD@mirror{%
  \fill
    node[anchor=south west,
         inner sep=0pt,
         yslant=\ProductBox@p@front@yslant,
         xscale=\ProductBox@p@front@xscale,
         yscale=-\ProductBox@p@mirror@yscale]
      {\begin{tikzpicture}
           \clip rectangle (\ProductBox@x,
                            \ProductBox@y*\ProductBox@p@mirror@yscale);
           \fill node [scope fading=ProductBoxFade,
                       yscale=\ProductBox@p@mirror@yscale,
                       inner sep=0pt]{\copy\ProductBox@Front};
```

```
          \end{tikzpicture}};
  \fill[xshift=-\ProductBox@p@left@xscale*\ProductBox@z,
        yshift=\ProductBox@p@top@yscale*\ProductBox@z]
    node[anchor=south west,
         inner sep=0pt,
         yslant=-\ProductBox@p@left@yslant,
         xscale=\ProductBox@p@left@xscale,
         yscale=-\ProductBox@p@mirror@yscale]
      {\begin{tikzpicture}
         \clip rectangle (\ProductBox@z,
                          \ProductBox@y*\ProductBox@p@mirror@yscale);
         \fill node [scope fading=ProductBoxFade,
                     yscale=\ProductBox@p@mirror@yscale,
                     inner sep=0pt]{\copy\ProductBox@Left};

       \end{tikzpicture}};
}
```

\ProductBox@threeD@flare This macro defines the code to produce the flare effect. The flare is achieved with overlaying a partially transparent and fading circle of white color.

```
\newcommand\ProductBox@threeD@flare{
  \fill [white,path fading=ProductBoxFlare]
    (.8*\ProductBox@x,.9*\ProductBox@y)
    circle(\ProductBox@flareDiameter);
}
```

The following fading is used for the flare effect in the 3D rendering.

```
\tikzfading[name=ProductBoxFlare,
            inner color=transparent!60,
            outer color=transparent!100]
```

Finally we define some variant names.

```
\@namedef{ProductBox@style@3D}{\ProductBox@style@threeD}
\@namedef{ProductBox@style@3d}{\ProductBox@style@threeD}
⟨/style⟩
```

That's all.

Change History

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; numbers in roman refer to the code lines where the entry is used.

R

S

T

U

X