

```
%META:TOPICINFO{author="HerbertVoss" date="1385565396" format="1.1" reprev="1.7"
version="1.7"}% %META:TOPICPARENT{name="WebHome"}%
```

Grafiken, Bilder, Farbe und andere Effekte

[Wie kann ich Grafiken in mein \(La\)TeX-Dokument einfügen?](#)
[Wie kann ich erreichen, daß anstelle der Grafiken nur leerer Platz in der Größe der Grafiken im LaTeX-Dokument erscheint?](#)
[Wie kann ich mit TeX farbigen Text setzen?](#)
[Bei mit ,XFig` erstellten und exportierten Zeichnungen erhalte ich bei der Einbindung in LaTeX den Fehler, dass "%%\SetFigFont%" nicht definiert ist. Was ist falsch?](#)
[Was kann ich tun, wenn eine Grafik im EPS-Format beim Einfügen in ein LaTeX-Dokument Probleme bereitet?](#)
[Warum werden Pixelgrafiken in pdf-Dateien, die ich mit pdfTeX erzeugt habe, verzerrt oder mit falschen Farben dargestellt?](#)
[Wie kann ich unter Windows Bilder, Grafiken etc. mit LaTeX verwenden?](#)
[Mit PiCTeX erhalte ich die Meldung , "%%No room for a new \dimen%"` . Wie kann ich das vermeiden?](#)
[Woher kann ich eine Dokumentation zum \(La\)TeX-Paket ,pictex` bekommen?](#)
[Wie kann ich eine Grafik unten rechts auf der Seite platzieren?](#)

[Wie kann ich Grafiken in mein \(La\)TeX-Dokument einfügen?](#)

Die LaTeX-Pakete `graphics` bzw. `graphicx` definieren die Anweisung `\includegraphics`, welche einen einfachen Weg darstellt, die Einbindung von Grafiken in ein LaTeX-Dokument in einer weitgehend systemunabhängigen Syntax vorzunehmen. Die früher für denselben Zweck verwendeten Anweisungen `\psfig`, `\epsfig`, `\epsfbox` usw. (enthalten in den Paketen `psfig`, `epsf` und `epsfig`) sollten nicht mehr verwendet werden, da sie teilweise fehlerhaft sind, nur für ausgewählte Treiber funktionieren und auch nicht weiterentwickelt werden.

Neben `\includegraphics` werden weitere nützliche Befehle zum Skalieren und Rotieren definiert. Die Größe der Grafik, die von LaTeX benötigt wird, um entsprechend Platz für die Grafik zu lassen, wird bei Encapsulated-PostScript-(EPS-)Dateien direkt aus dem ,BoundingBox`-Kommentar in der EPS-Datei gelesen. Bei anderen, insbesondere bei binär vorliegenden Grafikformaten wie ,gif` oder ,pcx` kann die Größe in einer begleitenden Textdatei bzw. im optionalen Argument der Anweisung `\includegraphics` angegeben werden. Mit dem Paketparameter `draft` des Paketes `graphics` kann man LaTeX veranlassen, in die dvi-Datei nur einen Rahmen in der Größe der Grafik statt des Verweises auf die Grafikdatei einzufügen, was das Betrachten beschleunigt und zum Korrekturlesen manchmal hilfreich ist.

Die Grafikinformation kann bei EPS-Dateien sowohl als Vektoren als auch in Form von Angaben zur Position der einzelnen Grafikpunkte (Pixelgrafik) enthalten sein. Wenn möglich, sollte man immer Vektorgrafiken bevorzugen, da sie ohne Qualitätsverluste skaliert werden können. Ist man, wie bei gescannten Bildern, auf Pixelgrafiken angewiesen, so sollte man zur Vermeidung von Qualitätsverlusten darauf achten, daß die Auflösung der Grafik derjenigen des Druckers entspricht. Man bedenke auch, daß Grautöne nur mit besser auflösenden Ausgabegeräten (mind. 600dpi, besser mehr) gut wiedergegeben werden.

Um vorhandene Pixelgrafiken nach EPS zu wandeln, kann man die Exportfunktion etlicher Grafikprogramme nutzen. Es gibt darüberhinaus die beiden Programme `,=jpeg2ps=`` und `,=tiff2ps=``,

die sehr kompakte EPS-Dateien erzeugen. Sie erfordern Ausgabegeräte, die ,PostScript Level 2` beherrschen, wie beispielsweise alle modernen PS-Drucker und neue Ghostscript-Versionen. Alternativ kann man zur Konvertierung von JPEG nach EPS auch ,imgtops2` verwenden, das in der EPS-Ausgabe DCT-Kompression nutzt und so sehr kompakte EPS-Dateien erzeugt. Zur Konvertierung von Pixelgrafiken, die im Format PNG, aber auch TIFF, JPEG und Netpbm vorliegen, in EPS für Geräte mit PostScript Level 1, 2 oder 3, gibt es das Programm ,bmeps`, für TIFF auch ,tif2eps`, das Ghostscript voraussetzt. Ein weiteres Programm, welches eine Vielzahl von Grafikformaten konvertieren kann, ist ,convert`, das Bestandteil der Programmsammlung ,ImageMagick` ist. Es steht ebenfalls für viele Betriebssysteme zur Verfügung. Da die meisten dvi-Preview-Programme enthaltene EPS-Grafiken nicht darstellen können, empfiehlt es sich, die dvi-Datei nach PostScript zu wandeln und dann ein PostScript-Preview-Programm (,gv`, ,gsview`, ,ghostview`, ,Ghostscript` etc.) zu verwenden.

Sehr ausführliche und gut verständliche Hinweise zu diesem Thema enthält Keith Reckdahls Artikel ,Using Imported Graphics in LaTeX2e` (epslatex) und das Buch ,The LaTeX Graphics Companion` von Michel Goossens u.a.

graphics/graphicx	macros/latex/required/graphicx/
graphicx (Plain-TeX)	macros/plain/graphics/
epslatex	info/epslatex.ps
	info/epslatex.pdf
Graphics and Colour with LaTeX:	http://www.linmpi.mpg.de/~daly/latex/grf.htm
LaTeX2e und Grafik:	http://www.loh-fi.de/howto/latex.htm
jpeg2ps	http://www.pdflib.com/jpeg2ps/
	nonfree/support/jpeg2ps/
imgtops2	http://www.cs.washington.edu/homes/dougz/postscript/imgtops2/
tiff2ps	graphics/tiff/
tif2eps	support/pstools/tif2eps/
bmeps	support/bmeps/
gif2eps	http://www.zib.de/Visual/software/
ImageMagick	http://www.imagemagick.org/

Wie kann ich erreichen, daß anstelle der Grafiken nur leerer Platz in der Größe der Grafiken im LaTeX-Dokument erscheint?

Mit der Option ,draft` der Pakete ,graphics` bzw. ,graphicx`

```
\usepackage[draft]{graphicx}
```

erreicht man, daß statt der Grafiken nur ein Rahmen in Größe der Grafik gezeichnet wird. Zusätzlich wird der Name der einzufügenden Datei in diesen Rahmen geschrieben.

Tip: Verwendet man die Option ,draft` in der Klassendeklaration, so gilt diese auch für das Paket ,graphicx`. Diese globale Option kann jedoch mit der Paketoption ,final` wieder überschrieben werden.

Will man weder Rahmen noch Dateinamen erhalten, so kann man das Paket ,crop` in der folgenden Weise nutzen:

```
\usepackage[nographics]{crop}[2001/10/07]
```

Den gegenteiligen Effekt erreicht man mit der Paketoption ‚notext‘.

crop [macros/latex/contrib/crop/](https://www.ctan.org/ctan/packages/macros/latex/contrib/crop/)

Wie kann ich mit TeX farbigen Text setzen?

Das LaTeX-Bundle ‚graphics‘ enthält das Paket ‚color‘ [](#), das Befehle zur Verfügung stellt, mit denen man Textteile einfärben, eine Box oder den Hintergrund einer Seite farbig hinterlegen kann. Dies funktioniert natürlich nur, wenn das Ausgabegerät (Drucker, Bildschirm) dazu in der Lage ist und der dvi-Treiber die dazu notwendige Funktionalität hat, entsprechende \special-Anweisungen mit den Farbbefehlen auszuwerten.

Um das Paket ‚color‘ an verschiedene Ausgabegeräte und dvi-Treiber anzupassen, muß eine entsprechende Definitionsdatei für den jeweiligen Treiber vorhanden sein bzw. noch erstellt werden. Auf jedem System sollte in der Datei ‚color.cfg‘ eine dvi-Treiber-Definition als Default angegeben sein. Das Paket ‚color‘ läßt sich dann wie folgt verwenden:

```
\usepackage{color}
```

Will man in einem Dokument einen vom Standard abweichenden dvi-Treiber verwenden, so kann man den Namen der dvi-Treiber-Definition als Paket-Option angeben:

```
\usepackage[dvips]{color}
```

erzeugt eine dvi-Datei mit Farb-\special-Anweisungen für den dvi-PostScript-Treiber ‚dvips‘. Danach kann man die entstehende dvi-Datei auch mit anderen dvi-Treibern übersetzen bzw. ausdrucken, jedoch werden in den meisten Fällen die Farb-\special-Anweisungen nicht erkannt und mit einer Warnung ignoriert.

Eine genauere Beschreibung der einzelnen Anweisungen und der vorhandenen Definitionsdateien findet man in der Anleitung ‚grfguide‘ im Paket ‚graphics‘.

graphics/color: [macros/latex/required/graphics/](https://www.ctan.org/ctan/packages/macros/latex/required/graphics/)

Bei mit ‚XFig‘ erstellten und exportierten Zeichnungen erhalte ich bei der Einbindung in LaTeX den Fehler, dass =\SetFigFont nicht definiert ist. Was ist falsch?

Das Programm ‚fig2dev‘ [](#), das XFig zum Wandeln und Exportieren von Fig-Bildern aufruft, hat in den Versionen vor Version 3.1pl1 einen Fehler, der dazu führt, daß das Makro \SetFigFont nicht immer definiert ist. Nach einem Update des Paketes ‚Transfig‘, welches ‚fig2dev‘ enthält, verschwindet der Fehler. Kann man auf den Update nicht warten, bietet sich auch als Einfachstlösung die Definition \newcommand{\SetFigFont}[3]{} in der Dokumentpräambel an.

transfig/fig2dev [graphics/transfig/](https://www.ctan.org/ctan/packages/graphics/transfig/)

Was kann ich tun, wenn eine Grafik im EPS-Format beim Einfügen in ein LaTeX-Dokument Probleme bereitet?

Nicht selten erzeugen Grafikprogramme beim Speichern der Ergebnisse im PostScript-Code keine oder eine fehlerhafte BoundingBox-Kommentarzeile. Diese Angabe ist jedoch beim Einfügen der Grafik in ein LaTeX-Dokument unbedingt erforderlich, da sonst der benötigte Platz nicht korrekt berechnet werden kann. Der im folgenden testweise eingefügte Rahmen zeigt die Größe der BoundingBox in Relation zur Grafik:

```
\usepackage{graphicx}
% ...
\begin{figure}
\setlength{\fboxsep}{-\fboxrule}
\fbox{\includegraphics{dateiname}}
\end{figure}
```

Ist die BoundingBox-Zeile fehlerhaft, kann folgendermaßen vorgegangen werden:

- Manuelle Korrektur:
 Man lade die EPS-Datei in einen PostScript-Viewer und notiere sich die Koordinaten der linken unteren (ll,lr) und der rechten oberen (ul,ur) Ecke des interessierenden Teils der Grafik, wobei die Maßeinheit bp (big points) verwendet werden muß. Diese Werte füge man mit Hilfe eines ASCII-Editors im Anschluß an die Zeile „%PS-Adobe ...“ am Anfang der EPS-Datei hinzu bzw. korrigiere die vorhandene Angabe entsprechend:
 `<verbatim>BoundingBox: ll lr ul ur </verbatim>` Statt des Eintrags in die EPS-Datei können die Werte auch als optionale Parameter der `\includegraphics` `<a name="76" id="76" title="76"></a>`-Anweisung angegeben werden: `<verbatim>\includegraphics[bb" ll lr ul ur]{dateiname} </verbatim>` * ***Automatische Korrektur:***
 Man verwende das Programm `epstool` in der folgenden Weise: `<verbatim>epstool -b -c -oausgabedatei.eps eingabedatei.eps </verbatim>` In der Datei `„eingabedatei.eps“` befinde sich die fehlerhafte Grafik, danach enthält `„ausgabedatei.eps“` die korrigierte Grafik. `epstool` erfordert die vorherige Installation des PostScript-Interpreters `Ghostscript`.
 Alternativ dazu kann versucht werden, das Problem unter Verwendung des Programms `epstopdf` zu lösen:
 `<p><verbatim>epstopdf --nogs --outfile=ausgabedatei.eps eingabedatei.eps </verbatim>`
 `<p>` Liegt die Grafik in Form einer einseitigen PostScript-Datei vor, kann man mit Hilfe des Programms `ps2eps` versuchen, diese in eine EPS-Datei mit korrekter BoundingBox-Angabe umzuwandeln: `<p><verbatim>ps2eps eingabedatei.ps </verbatim>` `ps2eps` erfordert ebenfalls das Programm `Ghostscript` sowie `perl`. Sollte trotz korrekter Angabe der BoundingBox ein ungünstiges Ergebnis entstehen, kann es daran liegen, daß die EPS-Datei fälschlicherweise PostScript-Anweisungen enthält, die auch den Bereich außerhalb der BoundingBox verändern. Mit der Option `[clip]` der Anweisung `\includegraphics` wird die Ausgabe auf den Bereich der BoundingBox beschränkt.
 `|"epstool/gsview"`
 `|ftp://support/ghostscript/ghostgum/ |"epstopdf" |http://www.tug.org/applications/pdftex/ |`
 `|"ps2eps" |[[http://mirror.ctan.org/support/ps2eps/support/ps2eps/]] |"Ghostscript (GPL)"`
 `|[[http://mirror.ctan.org/support/ghostscript/gnu/support/ghostscript/gnu/]] |"Ghostscript (AFPL)"|[[http://mirror.ctan.org/nonfree/support/ghostscript/aladdin/nonfree/support/ghostscript/aladdin/]]] ===== Warum werden Pixelgrafiken in pdf-Dateien, die ich mit pdfTeX erzeugt habe, verzerrt oder mit falschen Farben dargestellt? =====`
 Vermutlich ist die Ursache die veraltete fehlerhafte pdfTeX-Version 0.13d. Abhilfe schafft die Verwendung einer aktuellen Version (z.B. 1.49 oder neuer). Bei dieser Gelegenheit empfiehlt es sich, die im Zusammenhang mit dem LaTeX-Paket `graphicx` `<a name="78" id="78"></a>`

title="78"> wichtige Datei „pdfTeX.def“ und das LaTeX-Paket „hyperref“ ebenfalls zu erneuern. |"pdfTeX.def"|<http://www.tug.org/applications/pdftex/> | |"graphicx"|<http://mirror.ctan.org/pkg/graphicx/graphicx> | |"hyperref"|<http://mirror.ctan.org/pkg/hyperref/hyperref/> |] ===== Wie kann ich unter Windows Bilder, Grafiken etc. mit LaTeX verwenden? ===== Um Bilder u.ä. aus Windows-Programmen in LaTeX (nicht pdfLaTeX) einzubinden, sollte man diese im EPS-Format in eine Datei drucken. Da es mit den EPS-Dateien, die die Standard-EPS-Druckertreiber unter Windows erzeugen, immer wieder zu Problemen kommt, sollte man stattdessen die PostScript-Druckertreiber von Adobe installieren. Damit die EPS-Datei nun problemlos und korrekt unter LaTeX mit der Anweisung `\includegraphics` eingebunden werden kann, ist die Bounding-Box-Angabe in der Datei zu bestimmen und eventuell zu korrigieren (siehe [\[#75|Bounding Box\]](#)). Man sollte versuchen, immer mit "pdflatex" zu arbeiten, da dann die Grafiken im Format jpeg, png oder pdf verwendet werden können. ===== Wie kann ich eine Grafik unten rechts auf der Seite platzieren? =====

```
<code> \documentclass{article}
\usepackage[a4paper,margin=1in,showframe,marginparsep=0pt,marginparwidth=0pt]{geometry}
\usepackage{graphicx} \usepackage{eso-pic} \AddToShipoutPictureBG*{% nur auf der
ersten Seite \AtTextLowerLeft{\makebox[\textwidth][r]{%
\includegraphics[width=1cm,height=1cm,keepaspectratio]{example-image-a}}}}
\begin{document} Some text here \end{document} </code> |"eso-pic%"|eso-pic|
```

- Main.HerbertVoss - 16 Mar 2009

From:

<https://wiki.dante.de/> - **DanteWiki**

Permanent link:

<https://wiki.dante.de/doku.php?id=dantefaq:grafiken>

Last update: **2023/02/28 10:39**

