

# PostScript und PDF

## Ein Kurzbericht zum laufenden Projekt

Herbert Voß

11. September 2009

# Inhaltsverzeichnis

Einführung

Der damalige Ist-Zustand

Die Entscheidung für Perl

Schritt 1

Schritt 2

Schritt 3

Schritt 4

Optionen

PDF-Datei als Standardausgabe

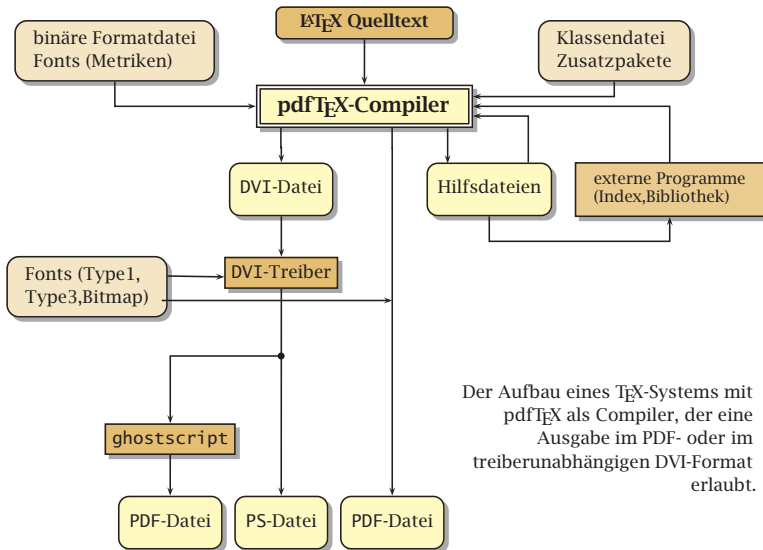
Erzeugte Dateien

Erzeugte Grafiken

Zukunft

Fragen

# Zur Erinnerung



Der Aufbau eines TeX-Systems mit pdfTeX als Compiler, der eine Ausgabe im PDF- oder im treiberunabhängigen DVI-Format erlaubt.

## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von `gex` (graphics extension) aus `VTEX` nach `pdfTEX`. Scheiterte an `micropress-inc`, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von `micropress-inc` sollte die Integration der Ghostscript-Bibliothek in den `pdfTEX`-Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des `TEX`-Quellcodes.
- Eine weitere Variante war der Einsatz von `LuaTEX`. Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren `LuaTEX` und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von `gex` (graphics extension) aus `VTEX` nach `pdfTEX`. Scheiterte an `micropress-inc`, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von `micropress-inc` sollte die Integration der Ghostscript-Bibliothek in den `pdfTEX`-Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des `TEX`-Quellcodes.
- Eine weitere Variante war der Einsatz von `LuaTEX`. Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren `LuaTEX` und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von `gex` (graphics extension) aus `VTEX` nach `pdfTEX`. Scheiterte an `micropress-inc`, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von `micropress-inc` sollte die Integration der Ghostscript-Bibliothek in den `pdfTEX`-Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des `TEX`-Quellcodes.
- Eine weitere Variante war der Einsatz von `LuaTEX`. Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren `LuaTEX` und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von `gex` (graphics extension) aus `VTEX` nach `pdfTEX`. Scheiterte an `micropress-inc`, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von `micropress-inc` sollte die Integration der Ghostscript-Bibliothek in den `pdfTEX`-Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des `TEX`-Quellcodes.
- Eine weitere Variante war der Einsatz von `LuaTEX`. Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren `LuaTEX` und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von gex (graphics extension) aus  $\text{V}\text{T}\text{E}\text{X}$  nach  $\text{pdf}\text{T}\text{E}\text{X}$ . Scheiterte an micropress-inc, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von micropress-inc sollte die Integration der Ghostscript-Bibliothek in den  $\text{pdf}\text{T}\text{E}\text{X}$ -Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des  $\text{T}\text{E}\text{X}$ -Quellcodes.
- Eine weitere Variante war der Einsatz von  $\text{Lua}\text{T}\text{E}\text{X}$ . Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren  $\text{Lua}\text{T}\text{E}\text{X}$  und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von `gex` (graphics extension) aus `VTEX` nach `pdfTEX`. Scheiterte an `micropress-inc`, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von `micropress-inc` sollte die Integration der Ghostscript-Bibliothek in den `pdfTEX`-Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des `TEX`-Quellcodes.
- Eine weitere Variante war der Einsatz von `LuaTEX`. Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren `LuaTEX` und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

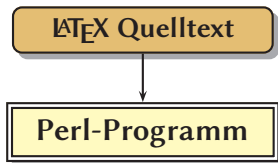
## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von `gex` (graphics extension) aus `VTEX` nach `pdfTEX`. Scheiterte an `micropress-inc`, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von `micropress-inc` sollte die Integration der Ghostscript-Bibliothek in den `pdfTEX`-Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des `TEX`-Quellcodes.
- Eine weitere Variante war der Einsatz von `LuaTEX`. Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren `LuaTEX` und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

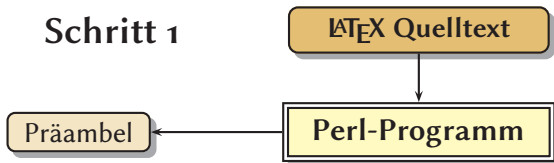
## Erste Schritte

- Die optimale Lösung wäre eine Übertragung von `gex` (graphics extension) aus `VTEX` nach `pdfTEX`. Scheiterte an `micropress-inc`, da sie Angst um ihre kommerzielle Windowsversion hatten.
- Nach Aussage von `micropress-inc` sollte die Integration der Ghostscript-Bibliothek in den `pdfTEX`-Code ohne Probleme möglich sein. Probleme bei der Übersetzung des Quellcodes zur Erzeugung der `gslib`. Lief eine Version, kam eine neue Ghostscriptversion und alles fing von vorne an ... Eine weitere Problematik war die historisch gewachsene ziemlich unübersichtliche Struktur des `TEX`-Quellcodes.
- Eine weitere Variante war der Einsatz von `LuaTEX`. Analog zur `METAPOST`-Bibliothek hätte man eine `PS`-Bibliothek aufbauen können. Zur damaligen Zeit waren `LuaTEX` und die `METAPOST-Lib` aber eher in einem dynamischen Zustand.

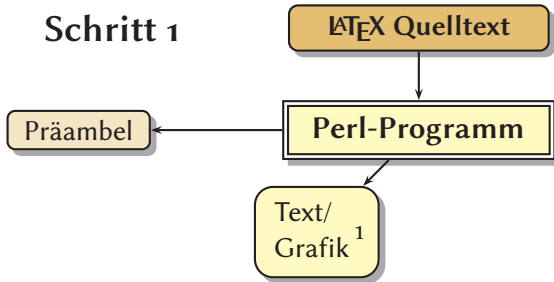
## Schritt 1



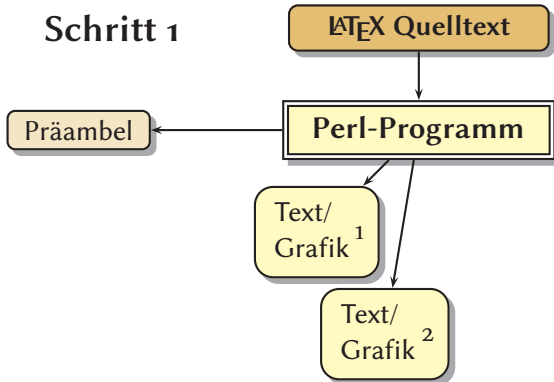
## Schritt 1



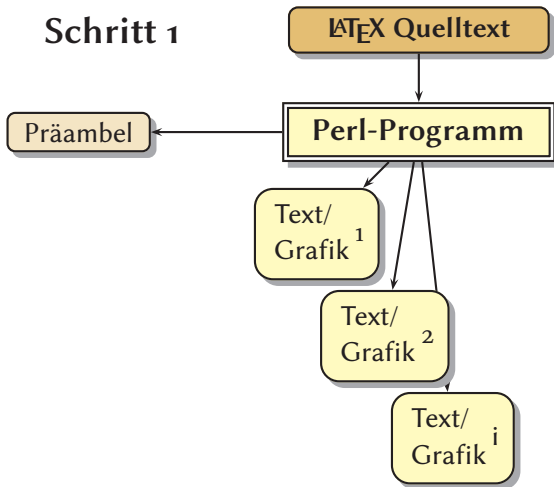
## Schritt 1



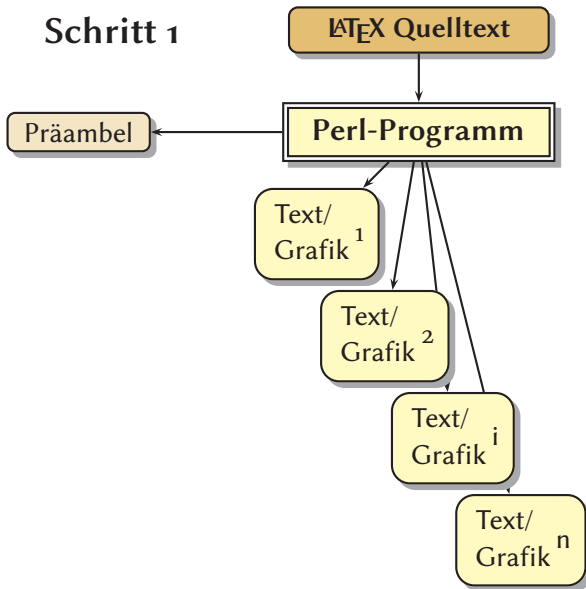
## Schritt 1



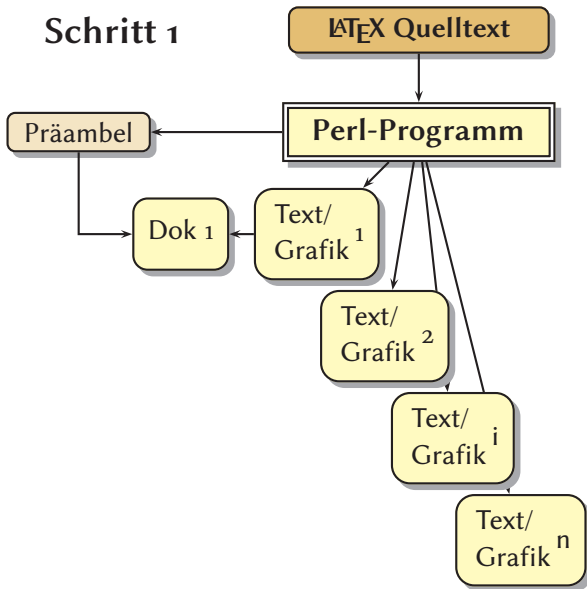
## Schritt 1



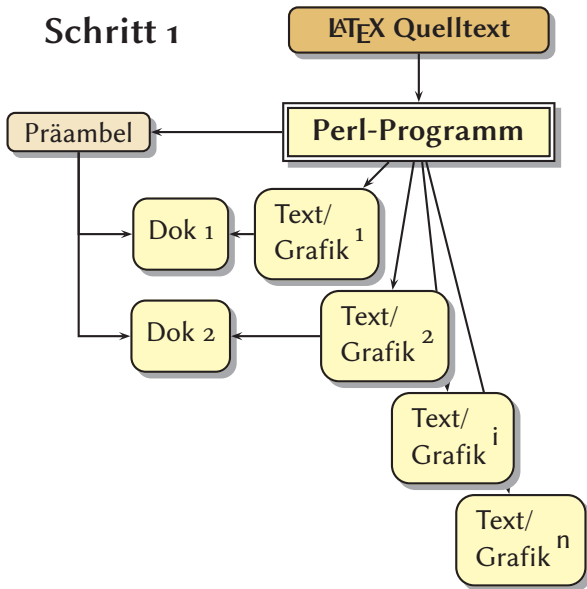
## Schritt 1



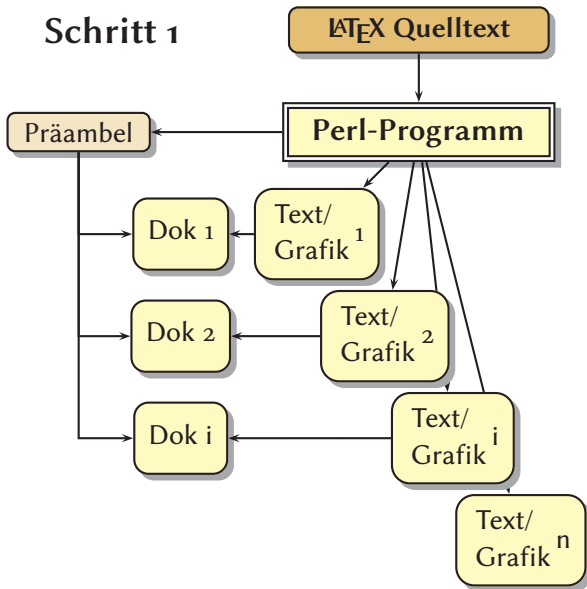
## Schritt 1



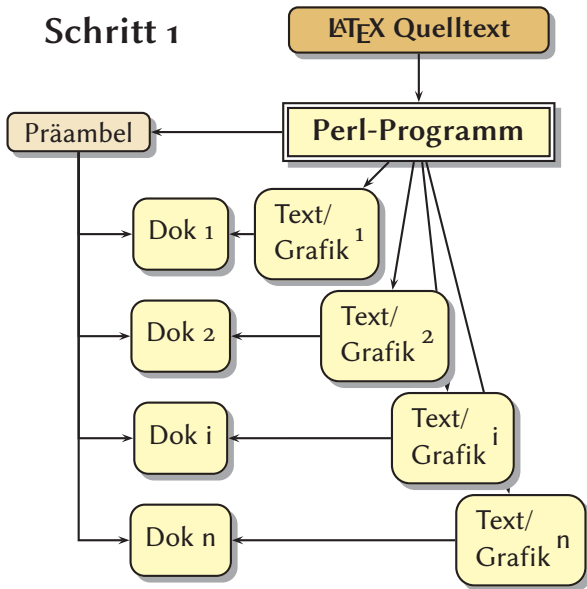
## Schritt 1



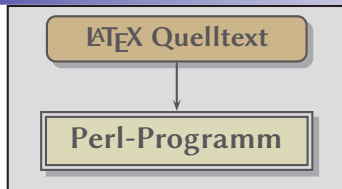
## Schritt 1



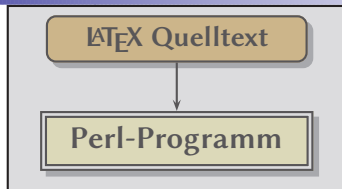
## Schritt 1



## Schritt 2

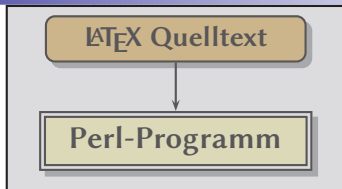
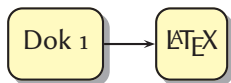


## Schritt 2

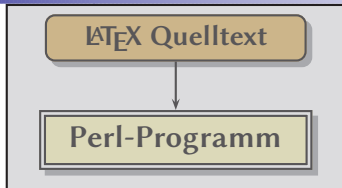


Dok 1

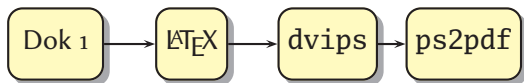
## Schritt 2



## Schritt 2



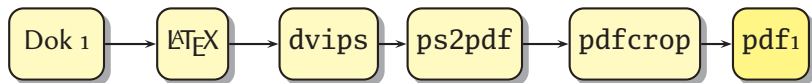
## Schritt 2



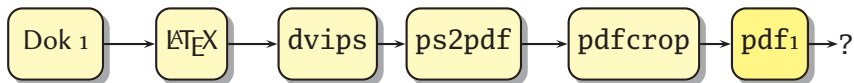
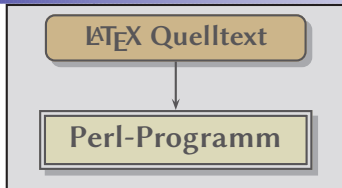
## Schritt 2



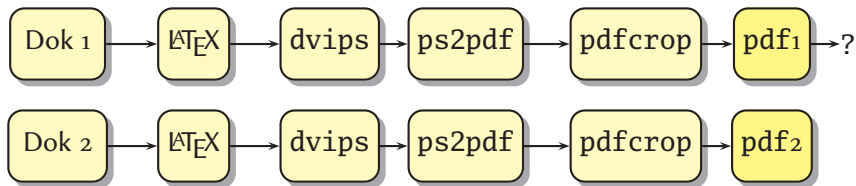
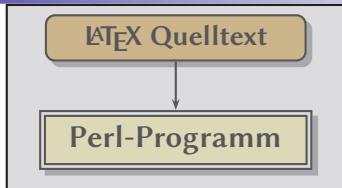
## Schritt 2



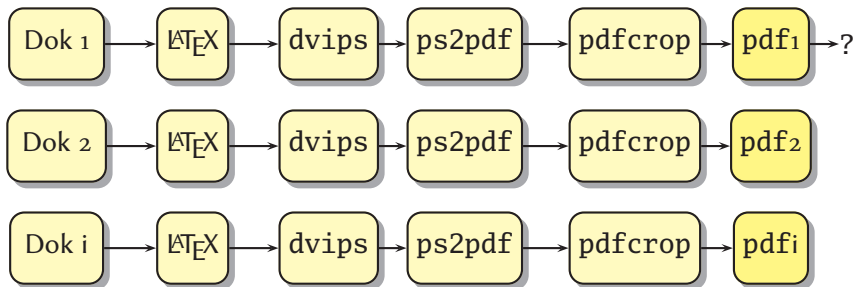
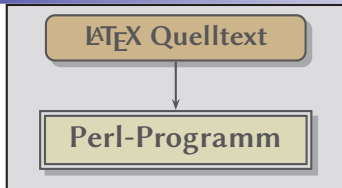
## Schritt 2



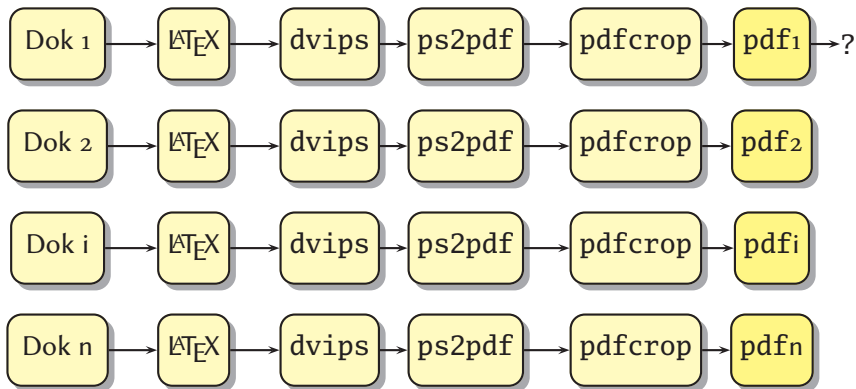
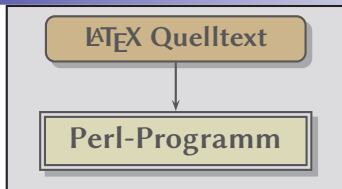
## Schritt 2



## Schritt 2



## Schritt 2



## Schritt 3

### ~~TeX~~ Quelltext

---

Normaler Text

```
\begin{postscript}
```

```
\includegraphics{bild.eps}
```

```
\end{postscript}
```

und weiter im Text

```
\begin{pspicture}(3,7)
```

```
...
```

```
\end{pspicture}
```

Wieder ganz normaler Text

```
...
```

## Schritt 3

### ~~TeX~~ Quelltext

---

Normaler Text

```
\begin{postscript}
\includegraphics{bild.eps}
\end{postscript}
```

und weiter im Text

```
\begin{pspicture}(3,7)
...
\end{pspicture}
```

Wieder ganz normaler Text

...

## Schritt 3

### ~~TeX~~ Quelltext

---

Normaler Text

```
\begin{postscript}
\includegraphics{bild.eps}
\end{postscript}
```

und weiter im Text

```
\begin{pspicture}(3,7)
...
\end{pspicture}
```

Wieder ganz normaler Text

...

## Schritt 3

### ~~TeX~~ Quelltext

---

Normaler Text

```
\includegraphics{bild1.pdf}
```

und weiter im Text

```
\begin{pspicture}(3,7)  
...  
\end{pspicture}
```

Wieder ganz normaler Text

...

## Schritt 3

### ~~La~~TeX Quelltext

---

Normaler Text

```
\includegraphics{bild1.pdf}
```

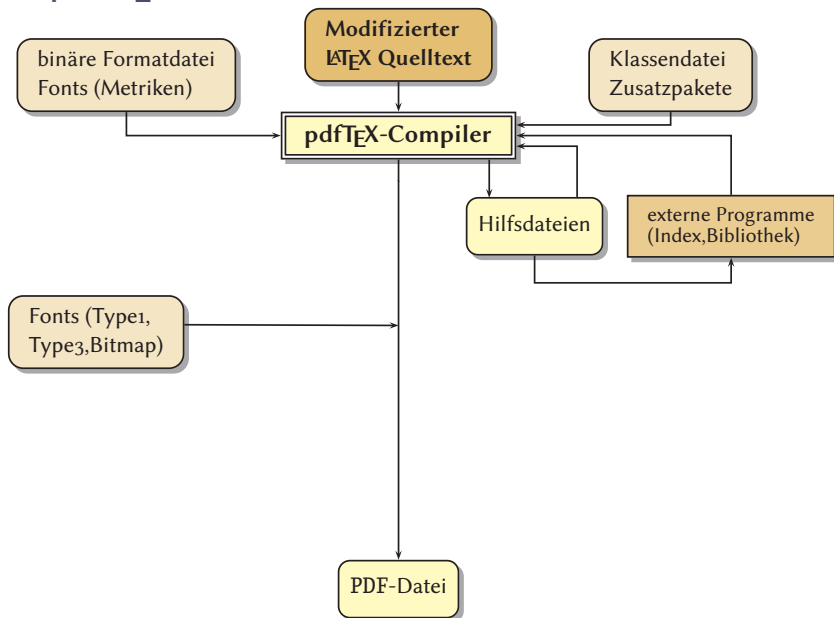
und weiter im Text

```
\includegraphics{bild2.pdf}
```

Wieder ganz normaler Text

...

# Der pdf $\LaTeX$ -Lauf



# Der pdf $\text{\LaTeX}$ -Lauf

```
#----- User part begin -----  
my $imageDir = "images";           # where to save the images  
my @imageType = ("eps","pdf","png"); # all image types (png only for  
my $Iext = ".png";                 # leave empty, if not a special  
my $tempDir = ".";                 # temporary directory  
my $verbose = 1;                   # 0 or 1, logfile  
my $clear = 0;                     # 0 or 1, clears all temporary  
my $DPI = 75;                      # very low value  
my $Iscale = 1;                    # for \includegraphics  
my $noImages = 0;                  # 1->create no images  
#----- User part end -----
```

# Grafiken

```
drwxr--r-- 2    4096 10. Sep 14:58 images
-rw-r--r-- 1    2234 10. Sep 14:57 pst-map2d-doc-pdf.aux
-rw-r--r-- 1    2871 10. Sep 14:57 pst-map2d-doc-pdf.bbl
-rw-r--r-- 1     935 10. Sep 14:57 pst-map2d-doc-pdf.blg
-rw-r--r-- 1   21164 10. Sep 14:57 pst-map2d-doc-pdf.log
-rw-r--r-- 1 1200123 10. Sep 14:57 pst-map2d-doc-pdf.pdf
-rw-r--r-- 1    8792 10. Sep 14:57 pst-map2d-doc-pdf.tex
-rw-r--r-- 1    1397 10. Sep 14:57 pst-map2d-doc-pdf.toc
-rw-r--r-- 1    7497 10. Sep 14:57 pst-map2d-doc.plog
-rw-r--r-- 1     848 10. Sep 14:56 pst-map2d-doc.preamble
-rw-r--r-- 1    9721 10. Sep 14:56 pst-map2d-doc.tex
-rw-r--r-- 1         8 10. Sep 14:57 pst-map2d-doc-tmp.aux
-rw-r--r-- 1    4560 10. Sep 14:57 pst-map2d-doc-tmp.dvi
-rw-r--r-- 1   10747 10. Sep 14:57 pst-map2d-doc-tmp.log
-rw-r--r-- 1  121815 10. Sep 14:57 pst-map2d-doc-tmp.pdf
-rw-r--r-- 1   88368 10. Sep 14:57 pst-map2d-doc-tmp.ps
-rw-r--r-- 1    1139 10. Sep 14:57 pst-map2d-doc-tmp.tex
-rw-r--r-- 1    5408 16. Mai 2008 PSTricks.bib
```

# Grafiken

```
-rw-r--r-- 1 296325 10. Sep 14:57 -000001.ppm
-rw-r--r-- 1 316202 10. Sep 14:56 pst-map2d-doc-tmp-0.pdf
-rw-r--r-- 1 44347 10. Sep 14:56 pst-map2d-doc-tmp-0.png
-rw-r--r-- 1 120940 10. Sep 14:57 pst-map2d-doc-tmp-10.pdf
-rw-r--r-- 1 46333 10. Sep 14:57 pst-map2d-doc-tmp-10.png
-rw-r--r-- 1 441616 10. Sep 14:56 pst-map2d-doc-tmp-1.pdf
-rw-r--r-- 1 119532 10. Sep 14:56 pst-map2d-doc-tmp-1.png
-rw-r--r-- 1 371700 10. Sep 14:57 pst-map2d-doc-tmp-2.pdf
-rw-r--r-- 1 49189 10. Sep 14:57 pst-map2d-doc-tmp-2.png
-rw-r--r-- 1 319201 10. Sep 14:57 pst-map2d-doc-tmp-3.pdf
-rw-r--r-- 1 47888 10. Sep 14:57 pst-map2d-doc-tmp-3.png
-rw-r--r-- 1 368020 10. Sep 14:57 pst-map2d-doc-tmp-4.pdf
-rw-r--r-- 1 63894 10. Sep 14:57 pst-map2d-doc-tmp-4.png
-rw-r--r-- 1 312167 10. Sep 14:57 pst-map2d-doc-tmp-5.pdf
-rw-r--r-- 1 95718 10. Sep 14:57 pst-map2d-doc-tmp-5.png
-rw-r--r-- 1 362826 10. Sep 14:57 pst-map2d-doc-tmp-6.pdf
-rw-r--r-- 1 58803 10. Sep 14:57 pst-map2d-doc-tmp-6.png
-rw-r--r-- 1 367190 10. Sep 14:57 pst-map2d-doc-tmp-7.pdf
-rw-r--r-- 1 65639 10. Sep 14:57 pst-map2d-doc-tmp-7.png
[...]
```

# Zukunft

- Nur noch eine einzige Umgebung PS, auf die getestet wird.
- Arbeiten mit Konfigurationsskripten, um weitere Grafikformate zu unterstützen.
- Eventuell grafisches Benutzerinterface erstellen.
- Test unter anderen Betriebssystemen.
- Integration in LuaTeX, um auf Perl verzichten zu können.

# Zukunft

- Nur noch eine einzige Umgebung PS, auf die getestet wird.
- Arbeiten mit Konfigurationsskripten, um weitere Grafikformate zu unterstützen.
- Eventuell grafisches Benutzerinterface erstellen.
- Test unter anderen Betriebssystemen.
- Integration in LuaTeX, um auf Perl verzichten zu können.

# Zukunft

- Nur noch eine einzige Umgebung PS, auf die getestet wird.
- Arbeiten mit Konfigurationsskripten, um weitere Grafikformate zu unterstützen.
- Eventuell grafisches Benutzerinterface erstellen.
- Test unter anderen Betriebssystemen.
- Integration in LuaTeX, um auf Perl verzichten zu können.

# Zukunft

- Nur noch eine einzige Umgebung PS, auf die getestet wird.
- Arbeiten mit Konfigurationsskripten, um weitere Grafikformate zu unterstützen.
- Eventuell grafisches Benutzerinterface erstellen.
- Test unter anderen Betriebssystemen.
- Integration in LuaTeX, um auf Perl verzichten zu können.

## Zukunft

- Nur noch eine einzige Umgebung PS, auf die getestet wird.
- Arbeiten mit Konfigurationsskripten, um weitere Grafikformate zu unterstützen.
- Eventuell grafisches Benutzerinterface erstellen.
- Test unter anderen Betriebssystemen.
- Integration in Lua $\text{T}\text{E}\text{X}$ , um auf Perl verzichten zu können.

