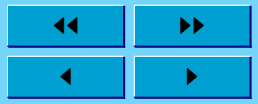
Using pdfTEX to build an industrial-strength PDF-based imposition program

Martin Schröder
ms@artcom-gmbh.de

EuroTEX 2001
September 24th–27th, 2001



1. An introduction to imposition

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)
Imposition means arranging pages on forms

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)
Imposition means arranging pages on forms

- Marks (small images) also are put on the forms for

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)
Imposition means arranging pages on forms

- Marks (small images) also are put on the forms for
  - Identification of jobs, pages and separations

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)
Imposition means arranging pages on forms

- Marks (small images) also are put on the forms for
  - Identification of jobs, pages and separations
  - Color proofing

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)
Imposition means arranging pages on forms

- Marks (small images) also are put on the forms for
  - Identification of jobs, pages and separations
  - Color proofing
  - Positioning of later steps like folding

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)
Imposition means arranging pages on forms

- Marks (small images) also are put on the forms for
  - Identification of jobs, pages and separations
  - Color proofing
  - Positioning of later steps like folding
- Objects can be rotated, normally by multiples of 90°

1. An introduction to imposition

Printing machines do not print single pages but forms that contain many pages (e. g. 8×20), meaning these forms are large (e. g. $2 \times 5\,\mathrm{m}$)
Imposition means arranging pages on forms

- Marks (small images) also are put on the forms for
  - Identification of jobs, pages and separations
  - Color proofing
  - Positioning of later steps like folding
- Objects can be rotated, normally by multiples of 90°
- „Botteling" involves rotating single objects by small degrees (e. g. 1°)

- Colored rectangles are generated by the software and can be placed on the forms

- Colored rectangles are generated by the software and can be placed on the forms
- In this case, color stands for CMYK or n channels (typically 8)

- Colored rectangles are generated by the software and can be placed on the forms
- In this case, color stands for CMYK or n channels (typically 8)
- As a result one large page contains a large number of both large and small objects

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images
  2. Assign pages to forms

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images
  2. Assign pages to forms
  3. Place marks

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images
  2. Assign pages to forms
  3. Place marks
  4. Make corrections

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images
  2. Assign pages to forms
  3. Place marks
  4. Make corrections
  5. Export (generate pdf)

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images
  2. Assign pages to forms
  3. Place marks
  4. Make corrections
  5. Export (generate pdf)
- Input is composite PDF, output is also composite PDF

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

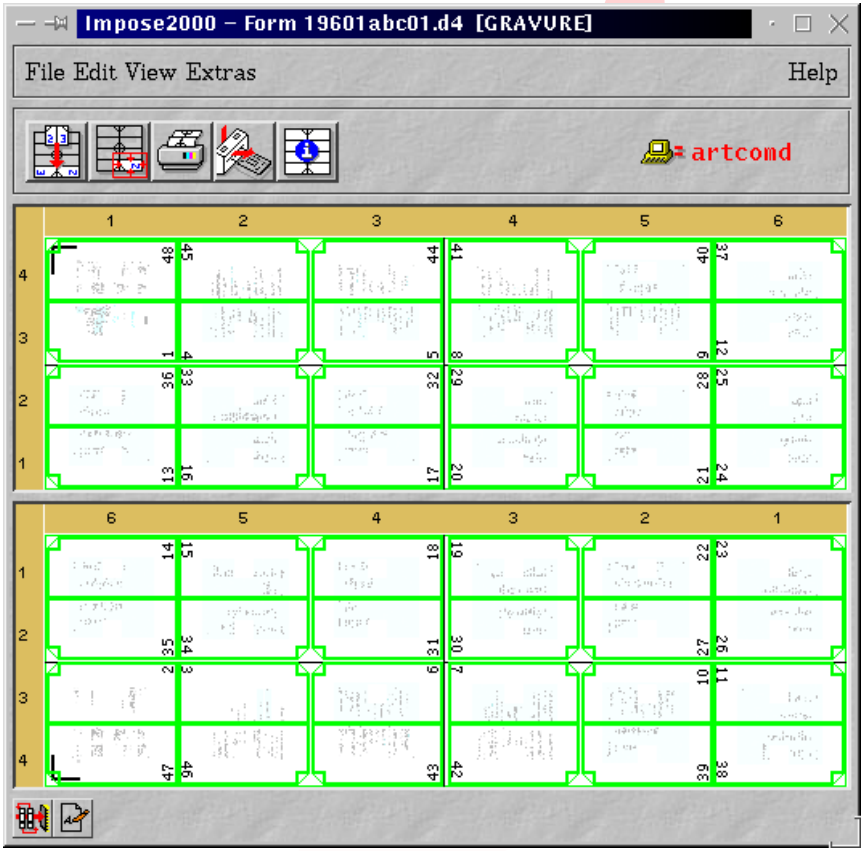
- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images
  2. Assign pages to forms
  3. Place marks
  4. Make corrections
  5. Export (generate pdf)
- Input is composite PDF, output is also composite PDF
- Separations are generated with a platesetter

2. Impose2000

ArtCom has developed an imposition program for PDF files: Impose2000.

- Runs under Unix (Linux, IRIX, Solaris) and X
- The workflow is:
  1. Generate overview images
  2. Assign pages to forms
  3. Place marks
  4. Make corrections
  5. Export (generate pdf)
- Input is composite PDF, output is also composite PDF
- Separations are generated with a platesetter
- Objects can be placed on specific separations

A screenshot with a form:



3. Why did we chose pdfTEX?

3. Why did we chose pdfTEX?

- We need an application for arranging and combining pages from PDF files

3. Why did we chose pdfTEX?

- We need an application for arranging and combining pages from PDF files
- There are very few libraries that can read PDF

3. Why did we chose pdfTEX?

- We need an application for arranging and combining pages from PDF files
- There are very few libraries that can read PDF
- Commercial libaries are expensive and closed-source

3. Why did we chose pdfTEX?

- We need an application for arranging and combining pages from PDF files
- There are very few libraries that can read PDF
- Commercial libaries are expensive and closed-source
- There is a free one in Java, but it is under GPL

3. Why did we chose pdfTEX?

- We need an application for arranging and combining pages from PDF files
- There are very few libraries that can read PDF
- Commercial libaries are expensive and closed-source
- There is a free one in Java, but it is under GPL
- pdfTEX is a script engine, which means the license (GPL) is no problem

3. Why did we chose pdfTEX?

- We need an application for arranging and combining pages from PDF files
- There are very few libraries that can read PDF
- Commercial libaries are expensive and closed-source
- There is a free one in Java, but it is under GPL
- pdfTEX is a script engine, which means the license (GPL) is no problem
- it is possible to produce arbitrary PDF-code with pdfTEX by using `\pdfliteral`

pdfTEX is the application we need, because it is

pdfTEX is the application we need, because it is

- portable

pdfTEX is the application we need, because it is

- portable
- free and open-source (GPL)

pdfTEX is the application we need, because it is

- portable
- free and open-source (GPL)
- bug-free

pdfTEX is the application we need, because it is

- portable
- free and open-source (GPL)
- bug-free
- stable

pdfTEX is the application we need, because it is

- portable
- free and open-source (GPL)
- bug-free
- stable
- scriptable

pdfTEX is the application we need, because it is

- portable
- free and open-source (GPL)
- bug-free
- stable
- scriptable
- a batch processor

pdfTEX is the application we need, because it is

- portable
- free and open-source (GPL)
- bug-free
- stable
- scriptable
- a batch processor
- and we have the necessary skills (LATEX and PDF) in-house

4. Our intermediate format: LDL

4. Our intermediate format: LDL

The information from the form-export, on which pdfTEX runs, is stored in an intermediate file, which is in our internal format: the Layout Description Language (LDL)

4. Our intermediate format: LDL

The information from the form-export, on which pdfTEX runs, is stored in an intermediate file, which is in our internal format: the Layout Description Language (LDL)

- An LDL describes one form

4. Our intermediate format: LDL

The information from the form-export, on which pdfTEX runs, is stored in an intermediate file, which is in our internal format: the Layout Description Language (LDL)

- An LDL describes one form
- The dimensions of the form are specified

4. Our intermediate format: LDL

The information from the form-export, on which pdfTEX runs, is stored in an intermediate file, which is in our internal format: the Layout Description Language (LDL)

- An LDL describes one form
- The dimensions of the form are specified
- For each object there is a transformation matrix and a clipping matrix

5. Transforming an LDL into LaTeX: LDL2PDF

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

- Input is an LDL

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

- Input is an LDL
- Output is LaTeX

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

- Input is an LDL
- Output is LaTeX
- The file names of images must be converted for TEX

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

- Input is an LDL
- Output is LaTeX
- The file names of images must be converted for TEX
- The base of the TEX-file is a header-file

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

- Input is an LDL
- Output is LaTeX
- The file names of images must be converted for TEX
- The base of the TEX-file is a header-file
- It is a header-file and not a package because

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

- Input is an LDL
- Output is LaTeX
- The file names of images must be converted for TEX
- The base of the TEX-file is a header-file
- It is a header-file and not a package because
  - it is a real header-file

5. Transforming an LDL into LaTeX: LDL2PDF

The LDL is not directly consumed by pdfTEX but instead a LaTeX file is generated which is processed by pdfTEX.

- Input is an LDL
- Output is LaTeX
- The file names of images must be converted for TEX
- The base of the TEX-file is a header-file
- It is a header-file and not a package because
  - it is a real header-file
  - it is easier to maintain than a package (it is at a fixed position relative to the the binary and we do not need to hassle with kpathsea)

- The „user”-part of the generated file

- The „user”-part of the generated file
  - is LaTeX

- The „user”-part of the generated file
  - is LaTeX
  - uses logical markup

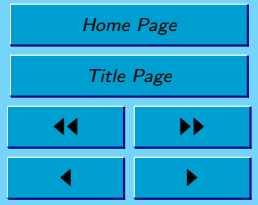
- The „user”-part of the generated file
  - is LaTeX
  - uses logical markup
  - uses the keyval-package so we can easily add and remove arguments in later releases

An example:

An example:

```
\ACGeometry{papersize={59.400000000cm,42.000000000cm}}
\begin{document}
\begin{ACForm}{formwidth=59.400000000,
               formheight=42.000000000,
               forminfo={/Creator (ArtCom I2K)
                         /Author (ArtCom I2K)
                         /Title (form1)}}
  \ACimage{%
    file=./T101794artcom8.pdf, pageno=8,
    imagetopage={0.000000000 29.703889062 ... },
    geotopage={0.000000000 21.000000000 ... },
    geotopageinvert={0.000000000 -0.033670034 ... },
  }%
  \ACimage{ ... }%
   ...
  \ACimage{ ... }%
\end{ACForm}
\end{document}
```

The implementation of these macros



The implementation of these macros

- is a mix of TEX and PDF

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling
- uses the geometry-package for setting the pagesize

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling
- uses the geometry-package for setting the pagesize
- handles options with the keyval-package

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling
- uses the geometry-package for setting the pagesize
- handles options with the keyval-package
- uses a `picture`-environment for fixing the position of objects

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling
- uses the geometry-package for setting the pagesize
- handles options with the keyval-package
- uses a `picture`-environment for fixing the position of objects
- uses `\pdfliteral` for constructing PDF-objects

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling
- uses the geometry-package for setting the pagesize
- handles options with the keyval-package
- uses a `picture`-environment for fixing the position of objects
- uses `\pdfliteral` for constructing PDF-objects
- allows the definition of separation colors

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling
- uses the geometry-package for setting the pagesize
- handles options with the keyval-package
- uses a `picture`-environment for fixing the position of objects
- uses `\pdfliteral` for constructing PDF-objects
- allows the definition of separation colors
- allows the generation of cropmarks, descriptions and labels

The implementation of these macros

- is a mix of TEX and PDF
- uses LATEX for the page-dimensions and option-handling
- uses the geometry-package for setting the pagesize
- handles options with the keyval-package
- uses a `picture`-environment for fixing the position of objects
- uses `\pdfliteral` for constructing PDF-objects
- allows the definition of separation colors
- allows the generation of cropmarks, descriptions and labels
- makes `\pdfimage` `\immediate` (this is needed because we have a large number of images on one page and normally `\pdfimage` keeps an image open till `\shipout`)

6. Setting up a TEX distribution

6. Setting up a TEX distribution

We have to distribute pdfTEX embedded into our software to systems where it will be installed, but we do not want to ship a TEXlive cd – all our software must fit on one cd

6. Setting up a TEX distribution

We have to distribute pdfTEX embedded into our software to systems where it will be installed, but we do not want to ship a TEXlive cd – all our software must fit on one cd

- We started from TEXlive

6. Setting up a TEX distribution

We have to distribute pdfTEX embedded into our software to systems where it will be installed, but we do not want to ship a TEXlive cd – all our software must fit on one cd

- We started from TEXlive
- We need binaries for Linux 2.0 (SuSE/Red Hat), IRIX 6.2 and Solaris-SPARC 2.5

6. Setting up a TEX distribution

We have to distribute pdfTEX embedded into our software to systems where it will be installed, but we do not want to ship a TEXlive cd – all our software must fit on one cd

- We started from TEXlive
- We need binaries for Linux 2.0 (SuSE/Red Hat), IRIX 6.2 and Solaris-SPARC 2.5
- The RPMs for SuSE are too old; teTEX is also too old (we need an up-to-date pdfTEX)

6. Setting up a TEX distribution

We have to distribute pdfTEX embedded into our software to systems where it will be installed, but we do not want to ship a TEXlive cd – all our software must fit on one cd

- We started from TEXlive
- We need binaries for Linux 2.0 (SuSE/Red Hat), IRIX 6.2 and Solaris-SPARC 2.5
- The RPMs for SuSE are too old; teTEX is also too old (we need an up-to-date pdfTEX)
- Since our platforms are old we have to generate the binaries by ourself

6. Setting up a TEX distribution

We have to distribute pdfTEX embedded into our software to systems where it will be installed, but we do not want to ship a TEXlive cd – all our software must fit on one cd

- We started from TEXlive
- We need binaries for Linux 2.0 (SuSE/Red Hat), IRIX 6.2 and Solaris-SPARC 2.5
- The RPMs for SuSE are too old; teTEX is also too old (we need an up-to-date pdfTEX)
- Since our platforms are old we have to generate the binaries by ourself
- The distribution should be as small as possible

- Our installation must not interfere with a given TEX-installation, but of course it has to be at a fixed place relative to the ArtCom-Software

- Our installation must not interfere with a given TEX-installation, but of course it has to be at a fixed place relative to the ArtCom-Software
- Our installation keeps its binaries at `bin/`, not `bin/$PLATTFORM/` – the ArtCom-installations are not multi-plattform

- Our installation must not interfere with a given TEX-installation, but of course it has to be at a fixed place relative to the ArtCom-Software
- Our installation keeps its binaries at `bin/`, not `bin/$PLATTFORM/` – the ArtCom-installations are not multi-plattform
- Our installation embeds no other PostScript-fonts but Computer Modern and the ones found in included files

- Our installation must not interfere with a given TEX-installation, but of course it has to be at a fixed place relative to the ArtCom-Software
- Our installation keeps its binaries at `bin/`, not `bin/$PLATTFORM/` – the ArtCom-installations are not multi-plattform
- Our installation embeds no other PostScript-fonts but Computer Modern and the ones found in included files
- We want to handle this distribution with CVS

7. Enhancing and fixing pdfTEX

7. Enhancing and fixing pdfTEX

We found some problems with pdfTEX, which we fixed or avoided:

7. Enhancing and fixing pdfTEX

We found some problems with pdfTEX, which we fixed or avoided:

- We tried the TIF-inclusion but found that it is only useful for grayscale or RGB images. We are dealing with CMYK or n-color images, so this feature is useless for us

7. Enhancing and fixing pdfTEX

We found some problems with pdfTEX, which we fixed or avoided:

- We tried the TIF-inclusion but found that it is only useful for grayscale or RGB images. We are dealing with CMYK or n-color images, so this feature is useless for us
- pdfTEX did not handle the `/Rotate` attribute of included images; it does now

7. Enhancing and fixing pdfTEX

We found some problems with pdfTEX, which we fixed or avoided:

- We tried the TIF-inclusion but found that it is only useful for grayscale or RGB images. We are dealing with CMYK or n-color images, so this feature is useless for us
- pdfTEX did not handle the `/Rotate` attribute of included images; it does now
- pdfTEX always used the `/CropBox` of included images, but our software expects the `/MediaBox`; you can now specify the pdf page box pdfTEX shall use

7. Enhancing and fixing pdfTEX

We found some problems with pdfTEX, which we fixed or avoided:

- We tried the TIF-inclusion but found that it is only useful for grayscale or RGB images. We are dealing with CMYK or n-color images, so this feature is useless for us
- pdfTEX did not handle the `/Rotate` attribute of included images; it does now
- pdfTEX always used the `/CropBox` of included images, but our software expects the `/MediaBox`; you can now specify the pdf page box pdfTEX shall use
- pdfTEX had some problems with keys in dictionaries in included pdfs; these are fixed

8. Tools we developed

8. Tools we developed

pdfTEX can not do everything we needed, so we developed some tools for internal use:

8. Tools we developed

pdfTEX can not do everything we needed, so we developed some tools for internal use:

- Marks and images that are not PDF must be converted to PDF. This is done either with the help of a RIP or by a Java-program

8. Tools we developed

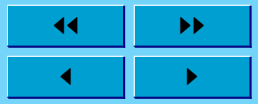
pdfTEX can not do everything we needed, so we developed some tools for internal use:

- Marks and images that are not PDF must be converted to PDF. This is done either with the help of a RIP or by a Java-program
- Color names and color spaces must be synchronized between all included images. This is done by a Java-program

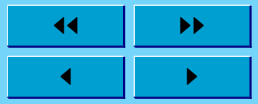
8. Tools we developed

pdfTEX can not do everything we needed, so we developed some tools for internal use:

- Marks and images that are not PDF must be converted to PDF. This is done either with the help of a RIP or by a Java-program
- Color names and color spaces must be synchronized between all included images. This is done by a Java-program
- Colored rectangles must be generated with n colors. This is done by a Java-program



9. Experience using pdfTEX



9. Experience using pdfTEX

- pdfTEX is fast

9. Experience using pdfTEX

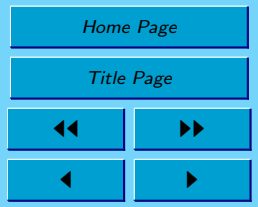
- pdfTEX is fast
- pdfTEX is reliable

9. Experience using pdfTEX

- pdfTEX is fast
- pdfTEX is reliable
- pdfTEX is extremly flexible thanks to `\pdfliteral`

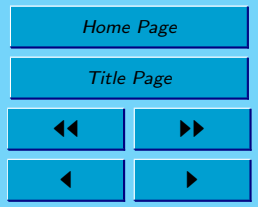
9. Experience using pdfTEX

- pdfTEX is fast
- pdfTEX is reliable
- pdfTEX is extremly flexible thanks to `\pdfliteral`
- The support for pdfTEX is incredibly good



9. Experience using pdfTEX

- pdfTEX is fast
- pdfTEX is reliable
- pdfTEX is extremly flexible thanks to `\pdfliteral`
- The support for pdfTEX is incredibly good
- Documenation for TEX is good; the system-documenation of pdfTEX could be better



9. Experience using pdfTEX

- pdfTEX is fast
- pdfTEX is reliable
- pdfTEX is extremly flexible thanks to `\pdfliteral`
- The support for pdfTEX is incredibly good
- Documenation for TEX is good; the system-documenation of pdfTEX could be better
- Hacking pdfTEX is complicated

9. Experience using pdfTEX

- pdfTEX is fast
- pdfTEX is reliable
- pdfTEX is extremly flexible thanks to `\pdfliteral`
- The support for pdfTEX is incredibly good
- Documenation for TEX is good; the system-documenation of pdfTEX could be better
- Hacking pdfTEX is complicated
- Importing pdfTEX into CVS is not trivial but can be done

9. Experience using pdfTeX

- pdfTeX is fast
- pdfTeX is reliable
- pdfTeX is extremly flexible thanks to `\pdfliteral`
- The support for pdfTeX is incredibly good
- Documenation for TeX is good; the system-documenation of pdfTeX could be better
- Hacking pdfTeX is complicated
- Importing pdfTeX into CVS is not trivial but can be done
- RIPs are a problem

9. Experience using pdfTEX

- pdfTEX is fast
- pdfTEX is reliable
- pdfTEX is extremly flexible thanks to `\pdfliteral`
- The support for pdfTEX is incredibly good
- Documenation for TEX is good; the system-documenation of pdfTEX could be better
- Hacking pdfTEX is complicated
- Importing pdfTEX into CVS is not trivial but can be done
- RIPs are a problem
- pdfTEX is a very good choice for creating and manipulating PDFs, especially in an industrial environment

10. Conclusion

10. Conclusion

Impose2000 is available for offset and gravure printing.

10. Conclusion

Impose2000 is available for offset and gravure printing.

Buy it! ☺