

MMDS

Introducing MMDS

What is MMDS

- MMDS is a software tool to transform text documents into some other form.
- In particular, it transforms plain text documents into nicely typeset printable documents.
- It uses the best typesetter there is: L^AT_EX.
- It can produce HTML (and several other formats) as well.
- Wait a minute... Aren't there a plethora of tools that do exactly that already?

Well, not quite

- Most plain text to nice formatted tools work by means of inserting markup commands in the plain text.

```
<h1>Plain text to HTML</h1>
<ul>
<li><p>Easy to use text.</p></li>
<li><p>Easy to use an &euml;. </p></li>
</ul>
```

```
\begin{document}
\section{Plain text to \LaTeX}
Easy to use text.\par
Easy to use an \"{e}.\par
```

MMDS is different

- MMDS tries to do its job with a minimal amount of user supplied markup.
- MMDS stands for: Minimal Markup Document System.
- It tries to infer from the user supplied text how it should be formatted.
- And yes, sometimes some markup is required for special effects.

It's Real!

- MMDS is not vaporware. It exists. It works. It is in use at several sites.
- It is a revamped version of the Squirrel Document System, developed during 1991-1994 for Multihouse Automatisering in Gouda.
- The first parts were written in Perl 3 (1991), the rest in Perl 4 (1992 and later).
- It stopped working with Perl 5.
- My fault – I accidentally made use of a Perl 4 bug that got fixed in Perl 5.

Current status

- It is used for notes, reports, meeting reports, offerings, and letters. And invoices. And presentations.
- But it is also used to typeset complete books, including graphics, table of contents, and index.

An example of MMDS input

Title: Example of MMDS input

Author: Johan Vromans

1. Using MMDS

Using MMDS is quite simple. Just type the input text using your favourite editor and make it look like it should.

- * Type text straightforward
- * Enumerations are simple as well

How does the output of this look?

Well, Like This

top of page

EXAMPLE OF MMDG INPUT

1. USING MMDG

Using MMDG is quite simple. Just type the input text using your favourite editor and make it look like it should.

- Type text straightforward
- Enumerations are simple as well

How does the output of this look?

bottom of page

JOHAN J. M. VROMANS

1

Design Goals

- Make it possible for everyone to produce high quality typeset documents
- Make it impossible to produce documents that do not adhere to the corporate guidelines

Make it possible for everyone ...

... to produce high quality typeset documents

- Only very few people can really use a word processor
- Only very few word processors can really produce high quality typeset documents
- It's not just people: software generates documents, too
- Using L^AT_EX directly is too difficult

Is L^AT_EX difficult

- It's not `\begin{document}` and friends that make it hard
- But secretaries want to type
 - ... according to a letter from mr. François, CEO of Deloitte & Touche, a price of \$625 (€598) is acceptable.
- And not
 - `\ldots{} according to a letter from mr.~Fran\,{c}ois, CEO of Deloitte \textampersand{} Touche, a price of \textdollar 625 (\euro 598) is acceptable.`
- Besides, you couldn't even do this with pre-2 ϵ

Make it impossible ...

- ... to produce documents that do not adhere to the corporate guidelines
- No matter how strict the guidelines are, people will find ways around them
- Templates are easily modified, sometimes accidentally
- People should put their energy in producing content, not layout
- MMDS was developed as part of a corporate quality project

Characteristics

- Basic input format is plain text. Input plugins can be supplied for different input formats (XML, POD, ...).
- Output is generated by output plugins. Plugins are provided for L^AT_EX, HTML and Text.
- Plugins for RTF and OpenOffice.org are under development.
- A driver program conveniently takes care of higher level conversions, like Text to PostScript or PDF.
- By default it produces PostScript and sends it to your system printer.
- It can also be used to preview any of the results.

Basic Principles

- A document is considered to consist of two parts: meta-data and content
- Meta-data describe the document and its properties, e.g., the document type, author, and title
- Meta-data are entered in RFC822-like form
- The meta-data section is terminated by a line of '-' or '=' characters

Title: Introduction to MMDS

Author: Johan Vromans

Version: X2.09

Contents

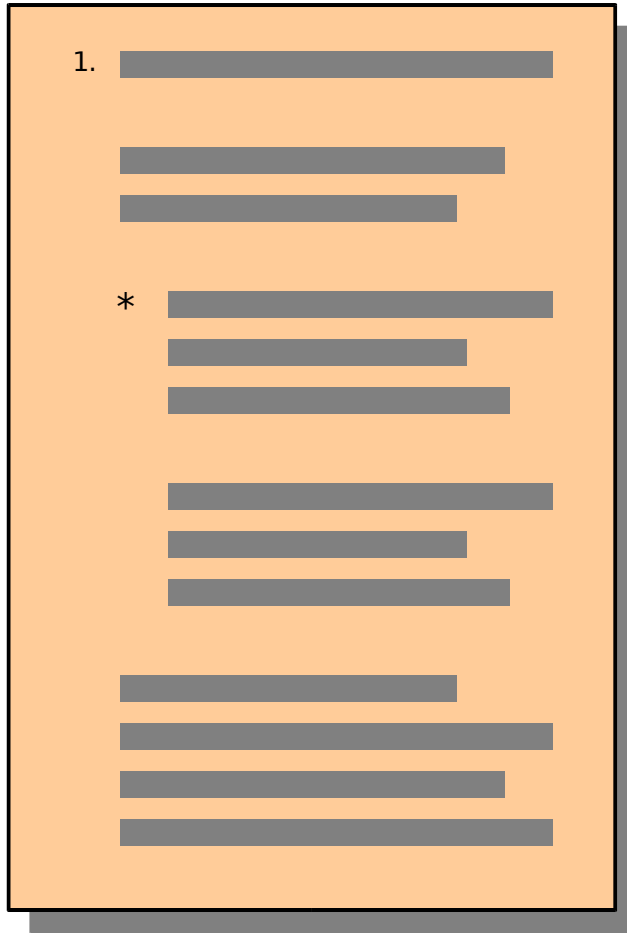
- The document contents consists of a series of paragraphs, separated by a blank line
- Paragraphs can be headers
 1. **Introduction**
- Text

In the beginning, there was no way ...
- Enumerations
 - * **The bright side**
 - **front**

Contents

- The actual type of paragraph is determined from the layout
- Leading space is significant
- Practice has shown that users do this correctly almost automatically
- A text editor that can do slightly intelligent word wrapping is a great help

Contents (cont'd)



heading

text

enumeration with bullet

enumeration cont'd

text

Document Types

- Memo's
- Meeting reports
- Technical Notes
- Reports
- Offerings
- Presentations (sheets)
- Letters

The document type is inferred from the meta-data. Explicit type selection is unnecessary (although possible)

Extensions

- Literal sections
- Aligning to columns
- Typefaces
- Images
- Tables
- In-lines
- Direct access to L^AT_EX

Literal sections

- For program code, listings and such

```
[literal]
```

```
This piece of text is rendered  
literally, and printed in a smaller,  
monospace font.
```

```
[end literal]
```

- Smaller variants: **screen** and **report**
- Option: **landscape**

Align to columns

- A simple yet powerful ‘ruler’ preceding a paragraph makes it possible to align columns

```
[      l      r      c      ]  
abc      def      ghi      jkl  
blahblah blahblah blahblah blahblah
```

- Comes out like this:

```
abc      def      ghi      jkl  
blahblah blahblah blahblah blahblah
```

Typefaces

- The original requirements explicitly specified that the user should not use bold, italic and so on
- The PostScript printers did not have the memory to contain that many downloaded fonts
- But technology advances
- For general discouragement, I invented something ugly: `@i@this will be italic@~i@`
- `i` \Rightarrow italic, `b` \Rightarrow bold, `t` \Rightarrow type, `u` \Rightarrow underline
- `f` \Rightarrow footnote
- This screws up the column alignment!

Images

- `[epsf image.eps]`
Includes the EPS graphic in the document
- `[epsf image.eps Descriptive title]`
Same, but with a nice descriptive title

Page breaks

- This turned out to be unavoidable...
 [`newpage`]
- Personally, I think users should not get involved with page breaks

Tables

- More ugly markup to support tables

```
[table 2cm,45mm,* type=3]
```

```
[row]
```

```
Cell 1/1 // Cell 2/1 // Cell 3/1
```

```
[row]
```

```
Cell 1/2 // Cell 2/2 // Cell 3/2
```

```
[end table]
```

- Each table cell can contain paragraphs, headers, everything except another table
- The type specifies the borders

In-lines

- `[inline tbl]`
For *tbl(1)* compatible tables
- `[inline screen]`
For nice screen images
- `[inline scalc]`
Reports from the SquirrelCalc Spreadsheet Calculator
- Most inlines have additional options, like `title`, `float`, `border`

Inline: tbl

```
[inline tbl title "Survey 2003"]
box center expand tab(#);
lrrc.
Year # In # Out # Profit

̄
2000 # 1200 # 1080 # Yes
2001 # 1210 # 1080 # Yes
2002 # 1220 # 970 # Yes
2003 # 960 # 970 # No
[end inline]
```

Inline: tbl (output)

Year	In	Out	Profit
2000	1200	1080	Yes
2001	1210	1080	Yes
2002	1220	970	Yes
2003	960	970	No

Survey 2003

Inline: screen

```
[inline screen expert title "Pop-up"]
!13x40

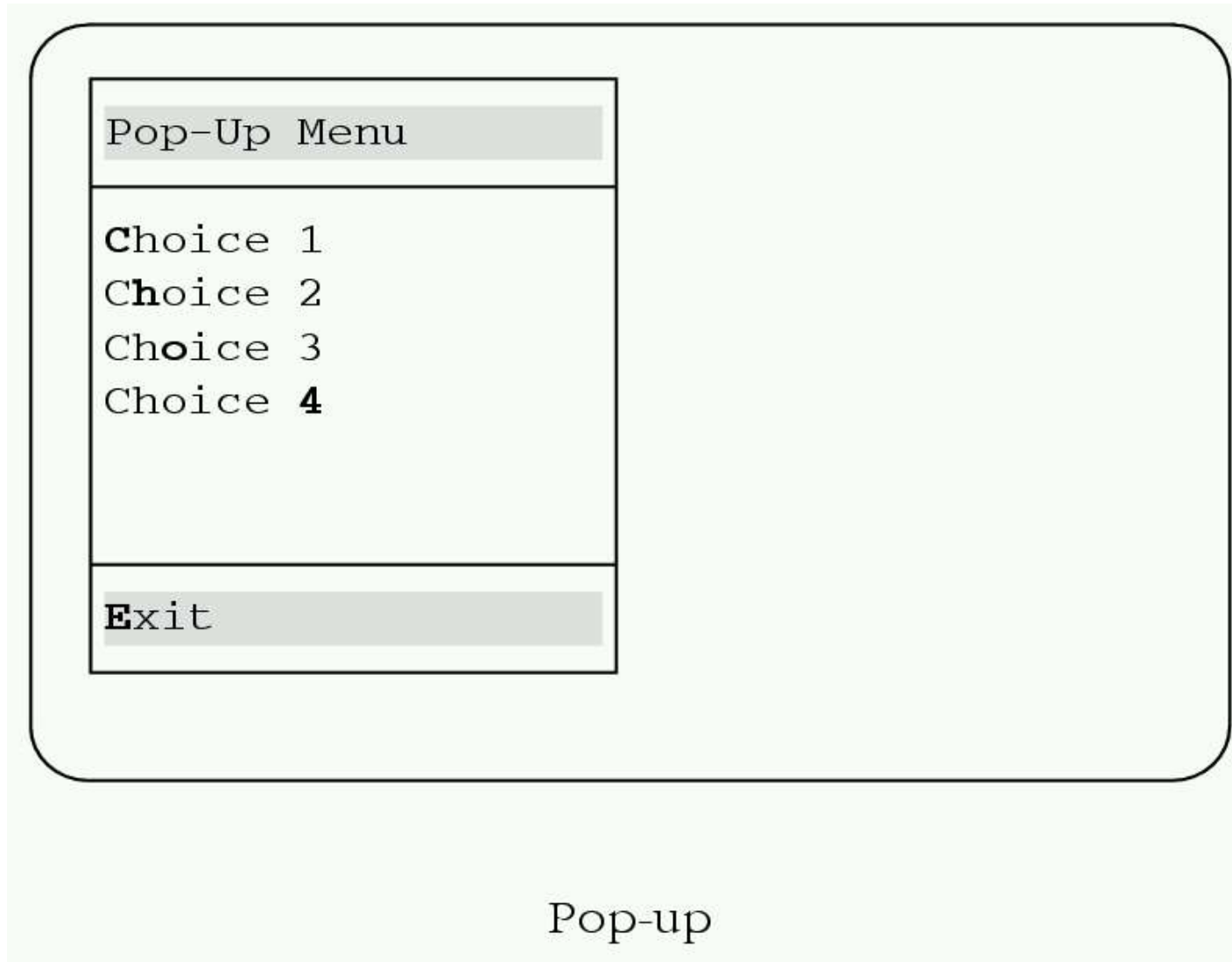
Pop-Up Menu

Choice 1
Choice 2
Choice 3
Choice 4

Exit

BX 1 1 +12 +20
HL 3 1 +20
HL 10 1 +20
R 2 2 +18
R 11 2 +18
B 4 2 +1
B 5 3 +1
B 6 4 +1
B 7 9 9
B 11 2 2
[end inline]
```

Inline: screen (output)



Inline: scalc

1. Inline scalc demo

A piece from a spreadsheet.

```
[scalc scalc.sc]
```

Neatly embedded.

```
r0c0 = [L] "Year" data: scalc.sc  
.....c0.....c1.....c2.....c3.....c4.....c5  
r0 Year In Out Profit  
r1 -----  
r2 2000 1200 1080 Yes  
r3 2001 1210 1080 Yes  
r4 2002 1220 970 Yes  
r5 2003 960 970 No  
r6  
r7  
r8  
r9  
r10  
r11
```

Inline: scalc (output)

1. `INLINE SCALC DEMO`

`A piece from a spreadsheet.`

<u>Year</u>	<u>In</u>	<u>Out</u>	<u>Profit</u>
2000	1200	1080	Yes
2001	1210	1080	Yes
2002	1220	970	Yes
2003	960	970	No

`Neatly embedded.`

Direct access to L^AT_EX

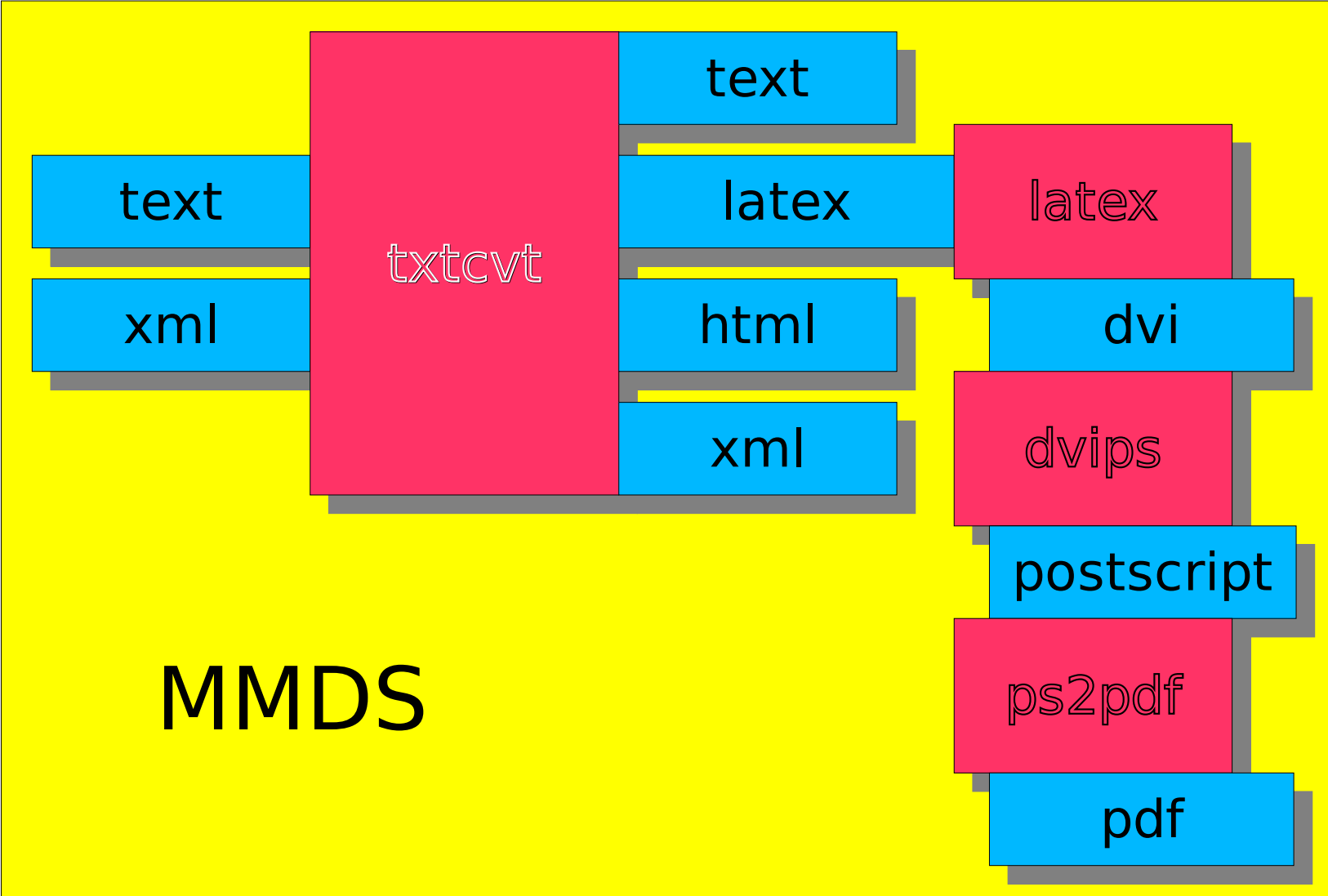
- Inserted unmodified into the L^AT_EX stream
- I implemented this mostly for my own convenience

```
[[TeX { \Large \center \vfill }]]
```

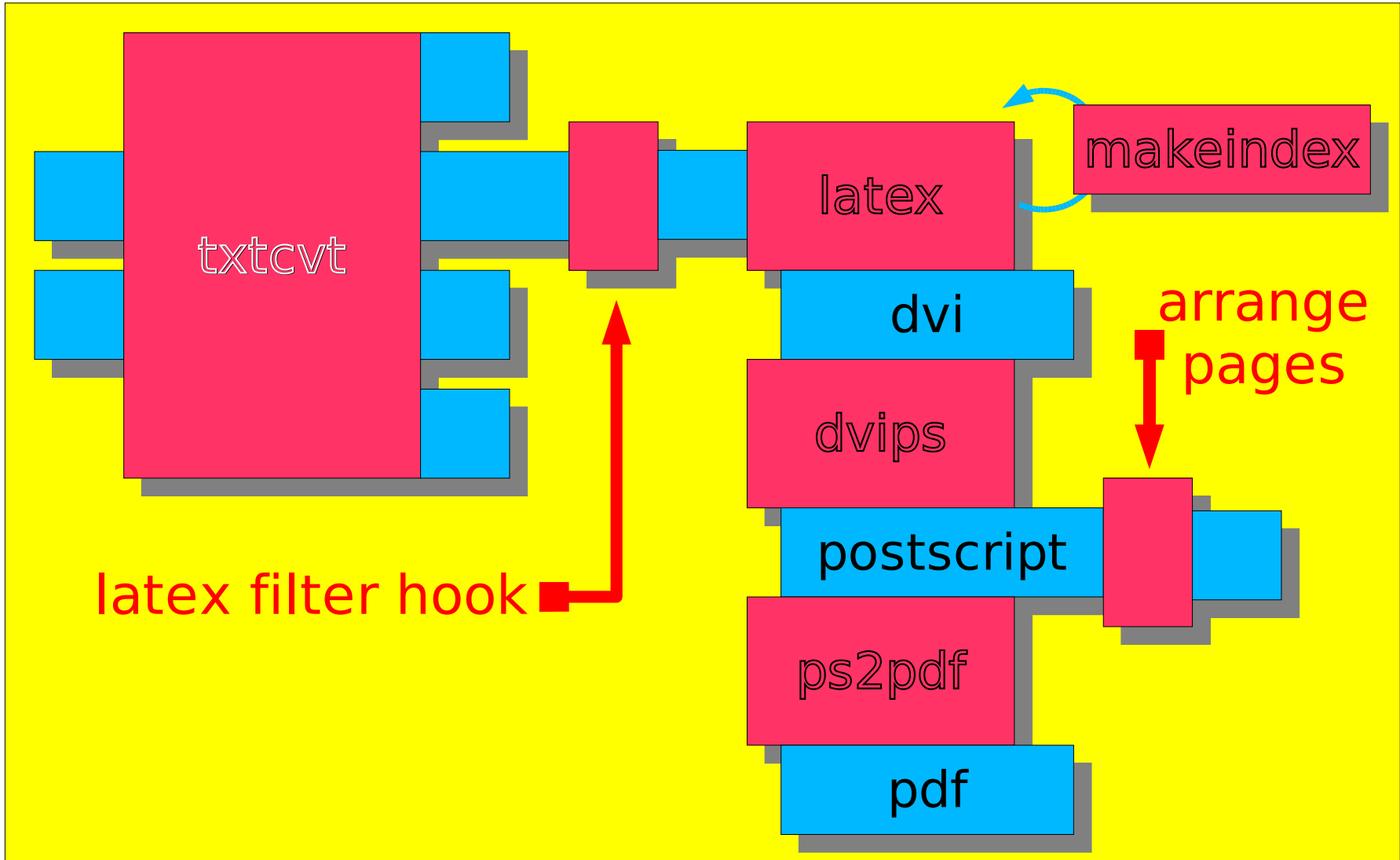
```
...
```

```
[[TeX \vfill } ]]
```

Package Overview



Power use



And next?

- MMDS will be made available for testing real soon.
- It will probably be located on SourceForge.
- With regular releases also on CPAN.

Why OpenOffice.org?

- Having a good tool does not mean there cannot be any others
- MMDS presentations are static
- They require a non-Open Source tool (Acrobat Reader) for presenting