

The ConTeXt presentation styles

a quick tour

What system ...



original

This is the oldest presentation style. Because it was used first for a large highly structured presentation, it comes with quite some screen real estate. Buttons as well as local tables of contents are rather dominant. The colors match the first screen version of the ConTeXt reference manual.

Meta Graphics

fun for non-wizzards

a demonstration



green

Because this style offers a lot of space, it is one of my favourites. The few simple buttons are typical METAPOST graphics. This style supports structuring. There are three main buttons and one extra button that can be customized.

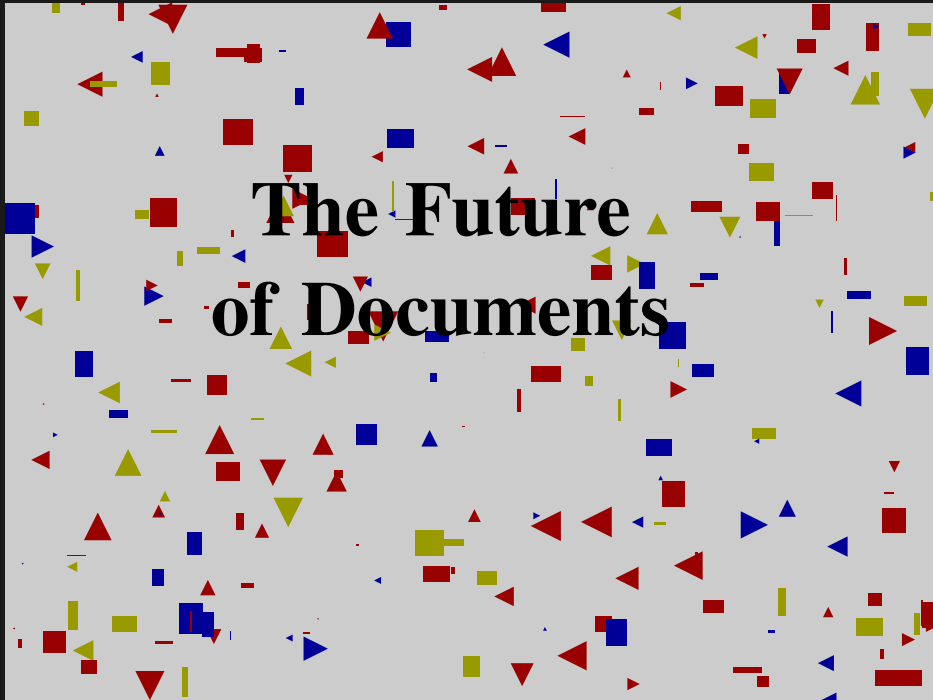
More than text alone

- graphics
- navigation
- fields
- intelligence

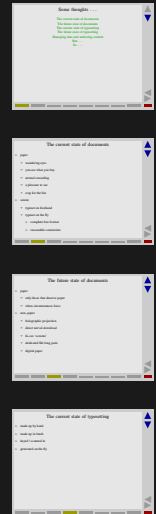


funny

This is a typical one-toc with just-a-few-pages style. The bottom half of the outline around the text shows the progress. This style is well suited for a summary from which one can launch other documents. This is how it was used first.



The Future of Documents

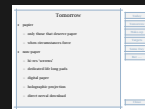


colorful

Although the title page of this style is rather random, the content can be well structure. There is a large text area available and the current position in the style is visualized at the bottom. The buttons are small METAPOST graphics.

Documents

some thoughts



fuzzy

ConTeXt users will recognize this style as being derived from the reference manual style. Indeed we use the same colors and random rectangles. At the right the main topics are shown and a close button. This style is well suited for itemized talks.

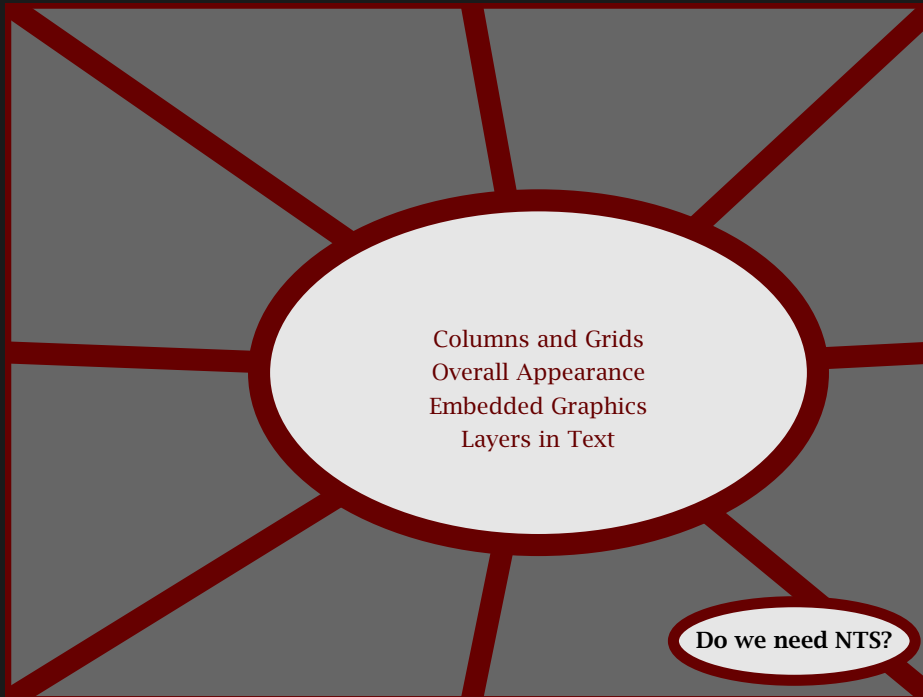
Think before you code

Quit



polish

The Antikwa Torunska font had a rather prominent backward slant. This style uses this font both for the running text and as background. The items follow the slant. This style is not meant for presentation with much text.



spider

In this style the text snippets are positioned at a random location on the page, to which the background graphic adapts itself. Each series of subtopics gets a different color. You can navigate the document by clicking on the circular shapes.

Fancy Graphics

positions and layers

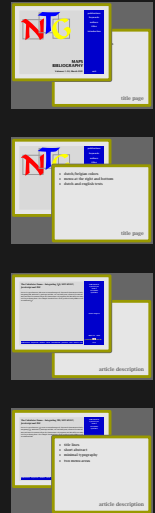


wonder

This is one of the 6 styles made for the nts presentation at EuroT_EX 1999. The idea was to demonstrate a couple of nasty things that one can do with pdfT_EX, being an example of an extension. Afterwards it was provided that this could also be done using traditional T_EX.

The MAPS Bibliography

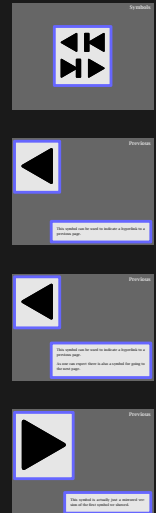
from XML to PDF



window

In order to explain the way the Maps bibliography was made, we needed a style to present both a sample and explanation.
This style presents pairs as well as single screens.

Some Famous Symbols



grow

This style looks like the previous one, but this time we build up the page. The dimensions of the text window are determined by the width of the sample window. As with its precursor, this style highly structures your talk.

Are we an endangered species?

(some thoughts and questions)

Hans Hagen

What we're going to discuss

- ✓ The situation
- ✓ The problem
- ✓ The solution
- ✓ The conclusion

Talk users

- ✓ The situation
- ✓ The problem
- ✓ The solution
- ✓ The conclusion

The publishers

- ✓ The situation
- ✓ The problem
- ✓ The solution
- ✓ The conclusion

The readers

- ✓ The situation
- ✓ The problem
- ✓ The solution
- ✓ The conclusion

writing

The Lucida Handwriting font is a pretty good font, and this style is meant to accompany it. The text frame is drawn by METAPOST using a penshape as are the small symbols that identify items and navigation buttons.



Facsimile Snapping Breaking Notes

1



split

This simple style can be used for relative short presentations with much text per screen. The colors are rather dark in order to let the text area shine bright.

The Future of NTS

Hans Hagen

The grand vision
of the future

The grand vision
of the future

The grand vision
of the future

The grand vision
of the future

balls

This is a special kind of style. You can stepwise build up a statement and present the individual steps as a piecewise built graphical paragraph. Each statement gets a different color.

Definitions

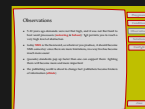
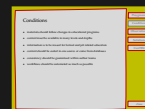
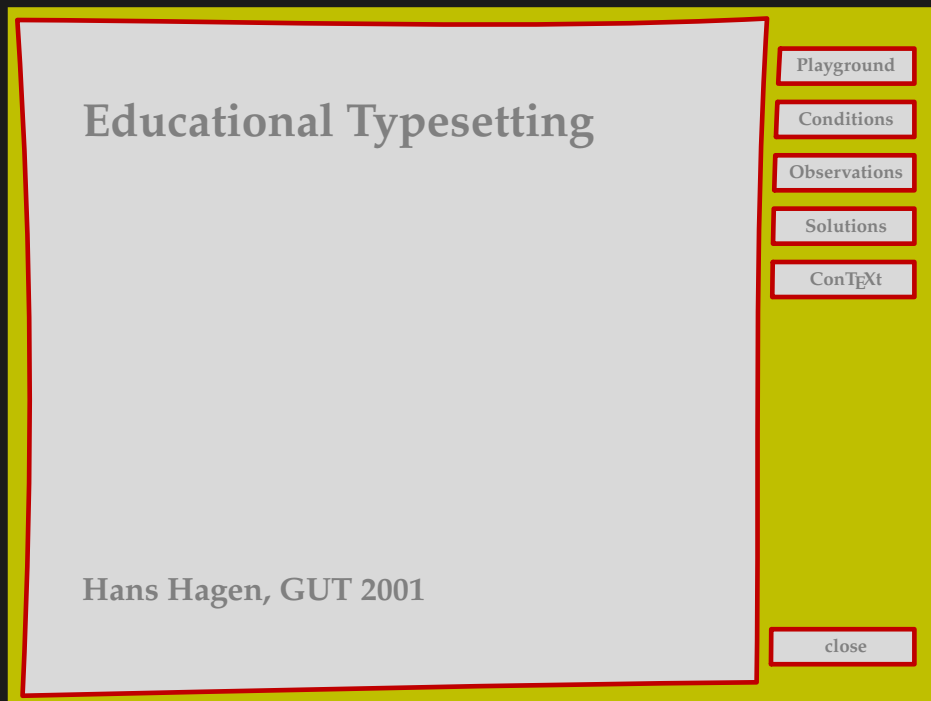
$\text{T}_{\text{E}}\text{X}$
 $\text{T}_{\text{E}}\text{X}$
MetaPost
Portable Document Format

JavaScript
eXtensible Markup Language
Con $\text{T}_{\text{E}}\text{X}$ t



knot

When you are in need of a quick and dirty presentation, this style can be your friend. It provides a minimum of structure and the scetchy outline add to the hasty job. Given the bit of randomness, it is also a typical example of a Con $\text{T}_{\text{E}}\text{X}$ t style.



organic

This style is more subtle than it looks at first sight. The buttons at the right, adapt themselves to the shape around the text, which can be a random one. This style was developed in the process of writing the MetaFun manual, and is described in detail in there. The outline comes in three flavors, one of them being a random shape.

Postprocessing PDF

It is not uncommon to postprocess the files

What is PDF

For long DVI was \TeX 's native output format. This format can be converted to for instance POSTSCRIPT or PDF. The later format has the advantage that fonts and graphics are embedded which make the file portable across platforms. We start this day with a short explanation of what PDF is.

Hans Hagen

The history of PDF

The PDF format was created by Adobe Systems Inc. in 1993. It was designed to be a portable document format that could be used to represent documents that could be printed or viewed on a computer. The format is based on the PostScript language, but it is more compact and easier to use. It is now the standard format for electronic documents.

From \TeX to PDF

The PDF format was created by Adobe Systems Inc. in 1993. It was designed to be a portable document format that could be used to represent documents that could be printed or viewed on a computer. The format is based on the PostScript language, but it is more compact and easier to use. It is now the standard format for electronic documents.

How PDF can improve your pages

The PDF format was created by Adobe Systems Inc. in 1993. It was designed to be a portable document format that could be used to represent documents that could be printed or viewed on a computer. The format is based on the PostScript language, but it is more compact and easier to use. It is now the standard format for electronic documents.

Graphics in PDF

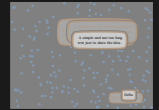
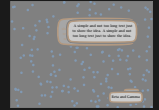
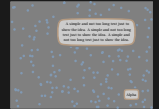
The PDF format was created by Adobe Systems Inc. in 1993. It was designed to be a portable document format that could be used to represent documents that could be printed or viewed on a computer. The format is based on the PostScript language, but it is more compact and easier to use. It is now the standard format for electronic documents.

cycle

This style was used at tug 2000 to introduce the pdf \TeX related talks. It cycles through the topics and summaries by moving them to the front one by one.

Something
Very Important

August 2000

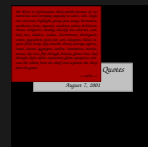


speckle

Sometimes a presentation is just a bunch of quotes. In that case this presentation style can be used to put them upfront. Each quote (or summary) is added to the previous page. By clicking on a quote one can go back to the page were it was presented. At the bottom we present the title.

The ConTEXt Test Quotes

August 7, 2001



more

This is another example of a style that stepwise builds up a screen. We cycle though the corners of the page with slightly random windows. It's one of my favourites.

The ConT_hxt Story

a quick tour

What output do ...
What input do ...
What T_hX do ...
What system ...

The next . . . minutes

What output do we want
What input do we prefer
What T_hX do we run
What system do we use

What output do ...
What input do ...
What T_hX do ...
What system ...

What output do we want

- high quality paper output
 - one input — multiple out-put
 - high level of abstraction
- optimal screen output
 - high quality typography
 - dedicated navigation
- some examples
 - local references
 - intelligent menus
 - text as program
 - lots of graphics
 - screen and paper
 - multiple fonts
 - simple but effective
 - B_oks and indices
 - cross linked source
 - typical T_hX
 - program as text

What output do ...
What input do ...
What T_hX do ...
What system ...

What input do we prefer

- highly structured ASCII
- medium neutral coding
- this file

What output do ...
What input do ...
What T_hX do ...
What system ...

What T_hX do we run

- DVI output
 - DVI generation
 - DVI viewing
- PDF output
 - pdf generation
 - pdf viewing Ghostscript
 - pdf viewer Acrobat

What output do ...
What input do ...
What T_hX do ...
What system ...

What system do we use

Starting point
Pitfalls
Structure
Typography
Navigation
User interface
Environment
Resources

What output do ...
What input do ...
What T_hX do ...
What system ...

Starting point

- typographic tradition
- eternal use
- optimal navigation
- future media
- new visions

What output do ...
What input do ...
What T_hX do ...
What system ...

Pitfalls

- structure
- fonts
- graphics
- compression
- portability
- searching
- interactivity
- quality
- flexibility
- protection

What output do ...
What input do ...
What T_hX do ...
What system ...

Structure

- multiple layouts
- reencoding of data
- degrees of freedom
- easy maintenance
- 255 file formats
- JavaScript
- sound and video

What output do ...
What input do ...
What T_hX do ...
What system ...

Typography

- nature (text, math, chem)
- character
- fragmentation
- floating bodies
- adaptive layout
- number substitutes
- enhanced pagbody
- color and backgrounds
- integrated METAPOST graphics

What output do ...
What input do ...
What T_hX do ...
What system ...

Navigation

- extensive sectioning
- local/global referencing
- long/cross references
- linked indices
- linked lists
- navigation bars
- subpage
- menus and buttons
- intelligence
- parallelism
- reader profiles
- version control

What output do ...
What input do ...
What T_hX do ...
What system ...

User interface

- parameter driven
- project organisation
- multi lingual
- minimal hacking needed
- hooks into main routines
- object oriented structure
- selective processing
- isolated specials

What output do ...
What input do ...
What T_hX do ...
What system ...

Environment

- programs
 - webl₂, emex, yandtex, pdfTeX
 - dvipson, GhostView, D_hatler, Reader
 - T_hXedit
- formats
 - Plain T_hX
 - ConT_hX

What output do ...
What input do ...
What T_hX do ...
What system ...

Resources

- documentation
- examples
- www.tex.ac.uk/contest
- mg.comcast.net/pd
- look force (Gilbert, Hans, Taro, Tobias)

What output do ...
What input do ...
What T_hX do ...
What system ...

original / pre-01

Meta Graphics
fun for non-wizzards
a demonstration

What I will discuss	
Including WinAPI code	Ticky race
Defining a Meta-Graphic	Logos
Resizing a graphic	Menu logos
Hooking into GDI	RTTI replacement
Communicating color	Bitmap graphics
Color conversions	Symbol sets
Graphic conversion / API's	Markers and annotations
Graphic conversions / API's	Colors and font
Primitives	Charts

[illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible]

```
Tricky runs

\startMFMrun
  \page_type := 212 ;
  \page_rh := 212 ;
  \pagef sep:=0pt ;
\stopMFMrun

\externalfigure{apure_212}[height=3cm]
```

Logos



◀ ▶

[illegible][illegible]

system	%	to R
Altair	10.4	JOE JOE!
MDIOG6	49.1	JOE JOE JOE JOE JOE JOE JOE JOE JOE JOE
OL/2	0.4	JOE JOE!
MacOS	5.7	JOE!
UNIX	11.9	JOE JOE JOE JOE JOE JOE JOE JOE JOE JOE
Windows	64.2	JOE JOE JOE JOE JOE JOE JOE JOE JOE JOE JOE

Symbol sets

```
\setkeys{key}{\showkeylabel{\navigation K}[n=4]}

\begin{tikzpicture}
    \draw[red] (0,0) --> (1,0);
    \draw[red] (1,0) --> (2,0);
    \draw[red] (2,0) --> (3,0);
    \draw[red] (3,0) --> (4,0);
    \draw[red] (4,0) --> (5,0);
    \draw[red] (5,0) --> (6,0);
    \draw[red] (6,0) --> (7,0);
    \draw[red] (7,0) --> (8,0);
    \draw[red] (8,0) --> (9,0);
    \draw[red] (9,0) --> (10,0);
    \draw[red] (10,0) --> (11,0);
    \draw[red] (11,0) --> (12,0);
    \draw[red] (12,0) --> (13,0);
    \draw[red] (13,0) --> (14,0);
    \draw[red] (14,0) --> (15,0);
    \draw[red] (15,0) --> (16,0);
    \draw[red] (16,0) --> (17,0);
    \draw[red] (17,0) --> (18,0);
    \draw[red] (18,0) --> (19,0);
    \draw[red] (19,0) --> (20,0);
    \draw[red] (20,0) --> (21,0);
    \draw[red] (21,0) --> (22,0);
    \draw[red] (22,0) --> (23,0);
    \draw[red] (23,0) --> (24,0);
    \draw[red] (24,0) --> (25,0);
    \draw[red] (25,0) --> (26,0);
    \draw[red] (26,0) --> (27,0);
    \draw[red] (27,0) --> (28,0);
    \draw[red] (28,0) --> (29,0);
    \draw[red] (29,0) --> (30,0);
    \draw[red] (30,0) --> (31,0);
    \draw[red] (31,0) --> (32,0);
    \draw[red] (32,0) --> (33,0);
    \draw[red] (33,0) --> (34,0);
    \draw[red] (34,0) --> (35,0);
    \draw[red] (35,0) --> (36,0);
    \draw[red] (36,0) --> (37,0);
    \draw[red] (37,0) --> (38,0);
    \draw[red] (38,0) --> (39,0);
    \draw[red] (39,0) --> (40,0);
    \draw[red] (40,0) --> (41,0);
    \draw[red] (41,0) --> (42,0);
    \draw[red] (42,0) --> (43,0);
    \draw[red] (43,0) --> (44,0);
    \draw[red] (44,0) --> (45,0);
    \draw[red] (45,0) --> (46,0);
    \draw[red] (46,0) --> (47,0);
    \draw[red] (47,0) --> (48,0);
    \draw[red] (48,0) --> (49,0);
    \draw[red] (49,0) --> (50,0);
    \draw[red] (50,0) --> (51,0);
    \draw[red] (51,0) --> (52,0);
    \draw[red] (52,0) --> (53,0);
    \draw[red] (53,0) --> (54,0);
    \draw[red] (54,0) --> (55,0);
    \draw[red] (55,0) --> (56,0);
    \draw[red] (56,0) --> (57,0);
    \draw[red] (57,0) --> (58,0);
    \draw[red] (58,0) --> (59,0);
    \draw[red] (59,0) --> (60,0);
    \draw[red] (60,0) --> (61,0);
    \draw[red] (61,0) --> (62,0);
    \draw[red] (62,0) --> (63,0);
    \draw[red] (63,0) --> (64,0);
    \draw[red] (64,0) --> (65,0);
    \draw[red] (65,0) --> (66,0);
    \draw[red] (66,0) --> (67,0);
    \draw[red] (67,0) --> (68,0);
    \draw[red] (68,0) --> (69,0);
    \draw[red] (69,0) --> (70,0);
    \draw[red] (70,0) --> (71,0);
    \draw[red] (71,0) --> (72,0);
    \draw[red] (72,0) --> (73,0);
    \draw[red] (73,0) --> (74,0);
    \draw[red] (74,0) --> (75,0);
    \draw[red] (75,0) --> (76,0);
    \draw[red] (76,0) --> (77,0);
    \draw[red] (77,0) --> (78,0);
    \draw[red] (78,0) --> (79,0);
    \draw[red] (79,0) --> (80,0);
    \draw[red] (80,0) --> (81,0);
    \draw[red] (81,0) --> (82,0);
    \draw[red] (82,0) --> (83,0);
    \draw[red] (83,0) --> (84,0);
    \draw[red] (84,0) --> (85,0);
    \draw[red] (85,0) --> (86,0);
    \draw[red] (86,0) --> (87,0);
    \draw[red] (87,0) --> (88,0);
    \draw[red] (88,0) --> (89,0);
    \draw[red] (89,0) --> (90,0);
    \draw[red] (90,0) --> (91,0);
    \draw[red] (91,0) --> (92,0);
    \draw[red] (92,0) --> (93,0);
    \draw[red] (93,0) --> (94,0);
    \draw[red] (94,0) --> (95,0);
    \draw[red] (95,0) --> (96,0);
    \draw[red] (96,0) --> (97,0);
    \draw[red] (97,0) --> (98,0);
    \draw[red] (98,0) --> (99,0);
    \draw[red] (99,0) --> (100,0);
    \draw[red] (100,0) --> (101,0);
    \draw[red] (101,0) --> (102,0);
    \draw[red] (102,0) --> (103,0);
    \draw[red] (103,0) --> (104,0);
    \draw[red] (104,0) --> (105,0);
    \draw[red] (105,0) --> (106,0);
    \draw[red] (106,0) --> (107,0);
    \draw[red] (107,0) --> (108,0);
    \draw[red] (108,0) --> (109,0);
    \draw[red] (109,0) --> (110,0);
    \draw[red] (110,0) --> (111,0);
    \draw[red] (111,0) --> (112,0);
    \draw[red] (112,0) --> (113,0);
    \draw[red] (113,0) --> (114,0);
    \draw[red] (114,0) --> (115,0);
    \draw[red] (115,0) --> (116,0);
    \draw[red] (116,0) --> (117,0);
    \draw[red] (117,0) --> (118,0);
    \draw[red] (118,0) --> (119,0);
    \draw[red] (119,0) --> (120,0);
    \draw[red] (120,0) --> (121,0);
    \draw[red] (121,0) --> (122,0);
    \draw[red] (122,0) --> (123,0);
    \draw[red] (123,0) --> (124,0);
    \draw[red] (124,0) --> (125,0);
    \draw[red] (125,0) --> (126,0);
    \draw[red] (126,0) --> (127,0);
    \draw[red] (127,0) --> (128,0);
    \draw[red] (128,0) --> (129,0);
    \draw[red] (129,0) --> (130,0);
    \draw[red] (130,0) --> (131,0);
    \draw[red] (131,0) --> (132,0);
    \draw[red] (132,0) --> (133,0);
    \draw[red] (133,0) --> (134,0);
    \draw[red] (134,0) --> (135,0);
    \draw[red] (135,0) --> (136,0);
    \draw[red] (136,0) --> (137,0);
    \draw[red] (137,0) --> (138,0);
    \draw[red] (138,0) --> (139,0);
    \draw[red] (139,0) --> (140,0);
    \draw[red] (140,0) --> (141,0);
    \draw[red] (141,0) --> (142,0);
    \draw[red] (142,0) --> (143,0);
    \draw[red] (143,0) --> (144,0);
    \draw[red] (144,0) --> (145,0);
    \draw[red] (145,0) --> (146,0);
    \draw[red] (146,0) --> (147,0);
    \draw[red] (147,0) --> (148,0);
    \draw[red] (148,0) --> (149,0);
    \draw[red] (149,0) --> (150,0);
    \draw[red] (150,0) --> (151,0);
    \draw[red] (151,0) --> (152,0);
    \draw[red] (152,0) --> (153,0);
    \draw[red] (153,0) --> (154,0);
    \draw[red] (154,0) --> (155,0);
    \draw[red] (155,0) --> (156,0);
    \draw[red] (156,0) --> (157,0);
    \draw[red] (157,0) --> (158,0);
    \draw[red] (158,0) --> (159,0);
    \draw[red] (159,0) --> (160,0);
    \draw[red] (160,0) --> (161,0);
    \draw[red] (161,0) --> (162,0);
    \draw[red] (162,0) --> (163,0);
    \draw[red] (163,0) --> (164,0);
    \draw[red] (164,0) --> (165,0);
    \draw[red] (165,0) --> (166,0);
    \draw[red] (166,0) --> (167,0);
    \draw[red] (167,0) --> (168,0);
    \draw[red] (168,0) --> (169,0);
    \draw[red] (169,0) --> (170,0
```

```
%define figurepanel [.....] [4p-cm-0]  
%atopfigurepanel
```

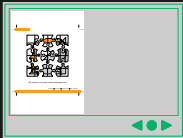
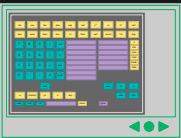
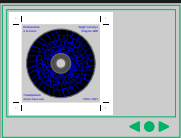
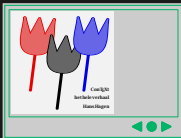
Covers and fun

```
\setexternalfigure  
[whatever]{page-300}  
[width=inner/laywidth,height=inner/layheight]  
\defineinnerlay  
[identifier]  
[{\externalfigure[whatever]}]  
\setpagebackgrounds  
[page]  
[background=identifier]
```



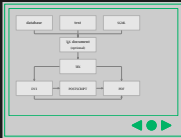
ConTeXt
commands

EN



Charts

```
\startFLMchart[process]
\startFLMcol
  \name {database}
  \location {1,1}
  \shape {action}
  \swnc {database}
  \connection [a] {source}
\stopFLMcol
...
\stopFLMchart
```



More than text alone

- graphics
- navigation
- fields
- intelligence

The calculator demo

Combining `gjs`, `gtkmm`, `gtk` and `javascript`.

Just a few examples

- in-line fill-in fields
- parents, children, clones and copies
- field characteristics
- entering via text
- all kinds of fields
- advanced references
- popping up information
- and some more
- figures and fields
- and more of those

Some new concepts

- reading objects
- selective inclusion
- one pass dilemma
- generating datastructures

Advanced referencing

Just some alternatives.

```
\goto[reference]  
\goto[outer reference:]  
\goto[outer reference:inner reference]  
\goto[operation(argument)]  
\goto[operation(action(argument,argument))]  
\goto[action]  
\goto[action(argument)]  
... chained or not.
```

funny / pre-03



Some thoughts . . .

- The current state of documents
- The future state of documents
- The current state of typesetting
- The future state of typesetting
- Managing data and authoring content
- But . . .
- So . . .

The current state of documents

- paper
 - wandering eyes
 - you see what you buy
 - eternal rewording
 - a pleasure to use
 - crap for the bin
- screen
 - typoist on keyboard
 - typoist on the fly
 - complete free format
 - reasonable constraints

The future state of documents

- paper
 - only those that deserve paper
 - when circumstances force
- new paper
 - holographic projection
 - direct neural download
 - hi-res 'screen'
 - dedicated life long path
 - digital paper

The current state of typesetting

- made up by hand
- made up in back
- keyed / scanned in
- generated on the fly

The future state of typesetting

- typesetting on demand
- intuitive authoring
- eternal reuse
- cheat watching

Managing data and authoring content

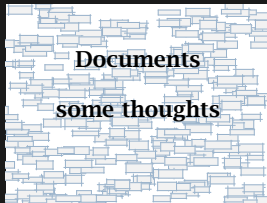
- what you think is what is stored
- what you think is what you get
- integrity testing / rebuilding
- author / user / design driven
- paid for giving and using
- personal storage / retrieval

But . . .

- there is nothing decent yet around
- we must make the best of it

So . . .

- why are we doing these things
- how to deal with the real looking
 - do we want to live with those temporary hacks
 - will I ever play a role in this
- when will the real genius stand up



Today

Tomorrow

Make-up

Targets

Some Day

But ...

Close

- paper
 - instant impressions
 - you see what you buy
 - can be a pleasure to see
 - more and more crap
- screen
 - typeset on forehand
 - typeset on the fly
 - complete free format
 - reasonable constraints

Today

Tomorrow

Make-up

Targets

Some Day

But ...

Close

Tomorrow

- papier
 - only those that deserve paper
 - when circumstances force
- non-paper
 - hi-res "screens"
 - dedicated life long pads
 - digital paper
 - holographic projection
 - direct neural download

Today

Tomorrow

Make-up

Targets

Some Day

But ...

Close

Make-up

- made up by hand
- made up automatically
- keyed and/or scanned in
- generated on the fly

Today

Tomorrow

Make-up

Targets

Some Day

But ...

Close

Targets

- intuitive authoring
- typesetting on demand
- eternal reuse
- clever searching

Today

Tomorrow

Make-up

Targets

Some Day

But ...

Close

Some Day

- thinking documents and content
- what you think is what's got stored
- what you think is what you get
- integrity testing / indexing
- author / user / design driven
- give and take, paid for or not
- personal life-long storage and retrieval

Today

Tomorrow

Make-up

Targets

Some Day

But ...

Close

But ...

- there is nothing decent yet around
- we must make the best of it
- what role can T&E play

fuzzy / pre-05

Think before you code

Quit

Topics

- document structure
- data abstraction
- optimal typography
- potential complications

Quit

Document structure

- provide non traditional sectioning
- use name spaced cross references
 - tables of contents must be able to adapt themselves"
 - anything can and will be reused"
 - many things can become a marginal, foot or endnote"
 - anything goes into headers, footers and sidebars"
 - there are more uses than we can think of

Quit

Data abstraction

- think of future demands and tools
- there is a difference between local and global data
- never code something more than once"
- force consistency by using abbreviations and aliases
- avoid hard coded names in changing documents"

Quit

Optimal typography

- it's in the small points, never compromise on kerning"
- tables can be typeset differently, flows can come back"
- math, physical units and chemicals are not the same
- languages have different demands and conventions
- we seldom know in advance what it will look"

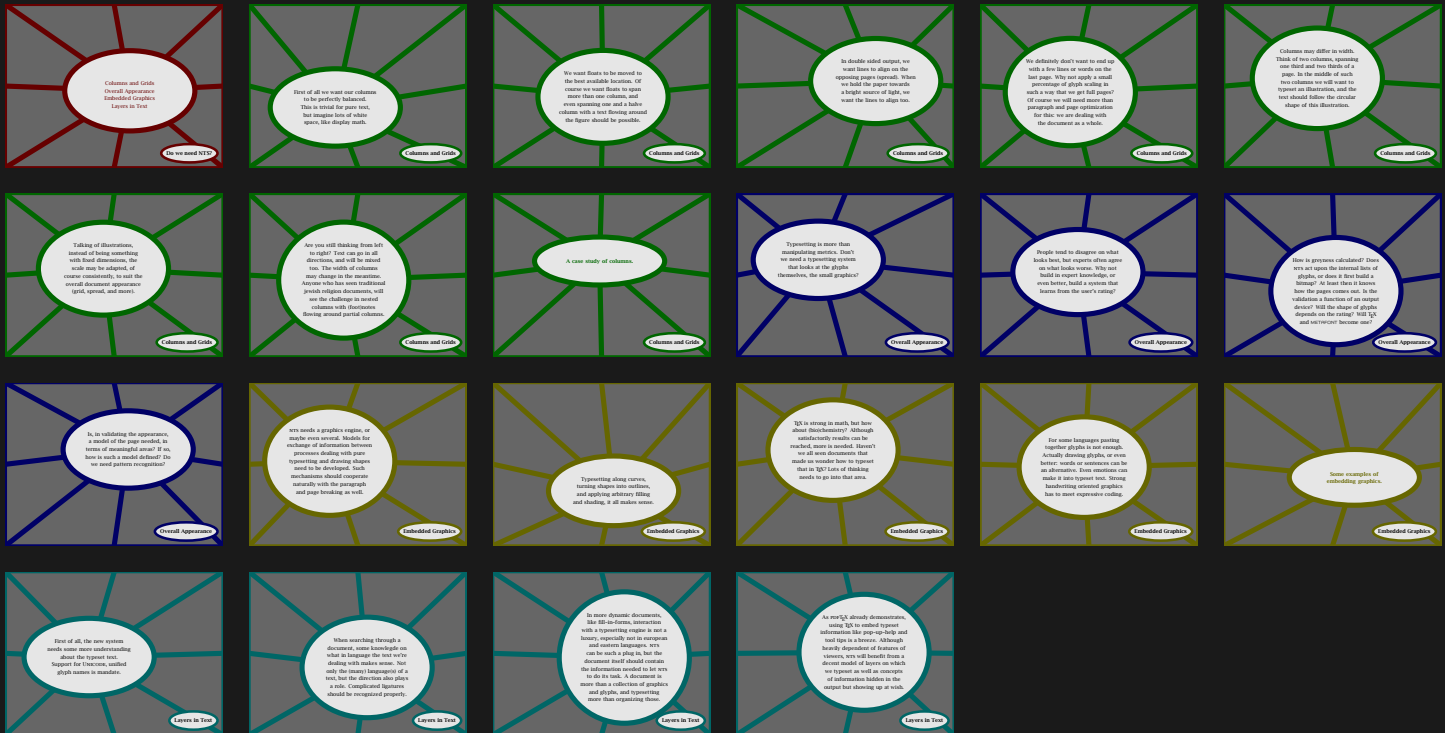
Quit

Potential complications

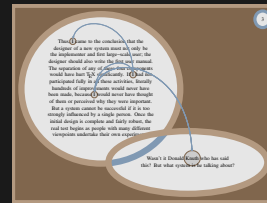
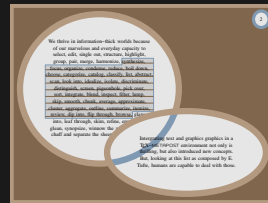
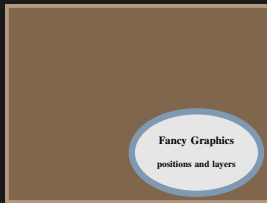
- lack of structure and consistency
- input, output and glyph encodings, forms and searching
 - use of periods, capitals, quotes and spaces"

Quit

polish / pre-06

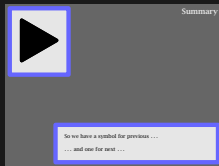
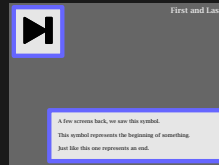
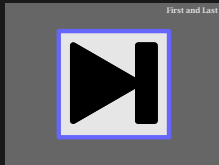
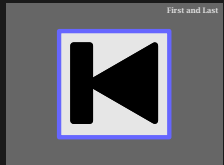
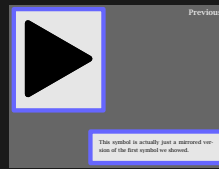
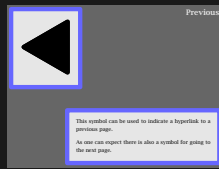
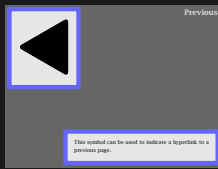
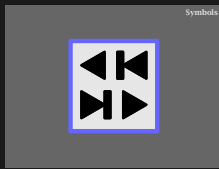
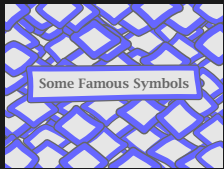


spider / pre-07



wonder / pre-08





Are we an endangered species?

(some thoughts and questions)

Hans Hagen

What we're going to discuss

TeX users
The publishers
The readers
Time
TeX and MetaPost
The future authors
The future publishers
The medium
TeX and MetaPost again
The coding
Conclusions

The users

- They are readers as well as authors.
- They often like writing.
- They have strong opinions on how things should look.
- Most of them are not trained in typesetting.
- Many of them are pretty well aware of how to save time.
- They like to concentrate on writing instead of formatting.
- They use TeX because of its quality.
- They can easily (and constructively) update their documents.
- They want to be (and often operate) at the front of technology.
- They don't know how to convince people to use TeX.

The publishers

- They used to consider typesetting to be a specialization.
- They are no longer primarily driven by content.
- They have to change, will change, and are changing.
- They want the highest quality for the lowest price.

The readers

- They move and move buy in WWW bookstores.
- They will start using these dedicated devices soon.
- Many of them still want to read books.
- They still prefer "look and feel over "style".
- They want to see in advance what they buy.
- They keep old copies of articles and books.
- They will finally get tired of updating.

Now ...

- will there be print?
- will there be design?
- will there be quality?
- what type of content will we deal with?
- who will be in control?

TeX and MetaPost

- Both can do a lot but are seldom used to full power.
- They are among the most powerful tools available.
- They demonstrated that typesetting can largely be automated.
- Like only, TeX and MetaPost have many fans.

The future authors

- They must learn to think in structure and reuse.
- Here and more they have to be aware of readers and presentation.
- They had better not easily accept that something cannot be done.
- They have to learn to think in mixed text and graphics.
- They should be aware of "interval" use and reuse.

The future publishers

- They will publish, but, as well as typset on demand.
- They will use techniques that go far behind current technologies.
- They will have to decide between good and bad (and quality).
- They have to combine psychology, ergonomics and technology.

The medium

- Are publishers willing to distribute document source code?
- How many help grids do we expect to find?
- What technologies can we expect?
- We will move from paper to screen to mind.

TeX and MetaPost again

- TeX is no easy job for TeX and MetaPost.
- In many areas they had competitors.
- They are still the most accurate tools available.
- Both provide cutting edge technology.
- Their users want to be in control.
- We still need to sort out some problems.

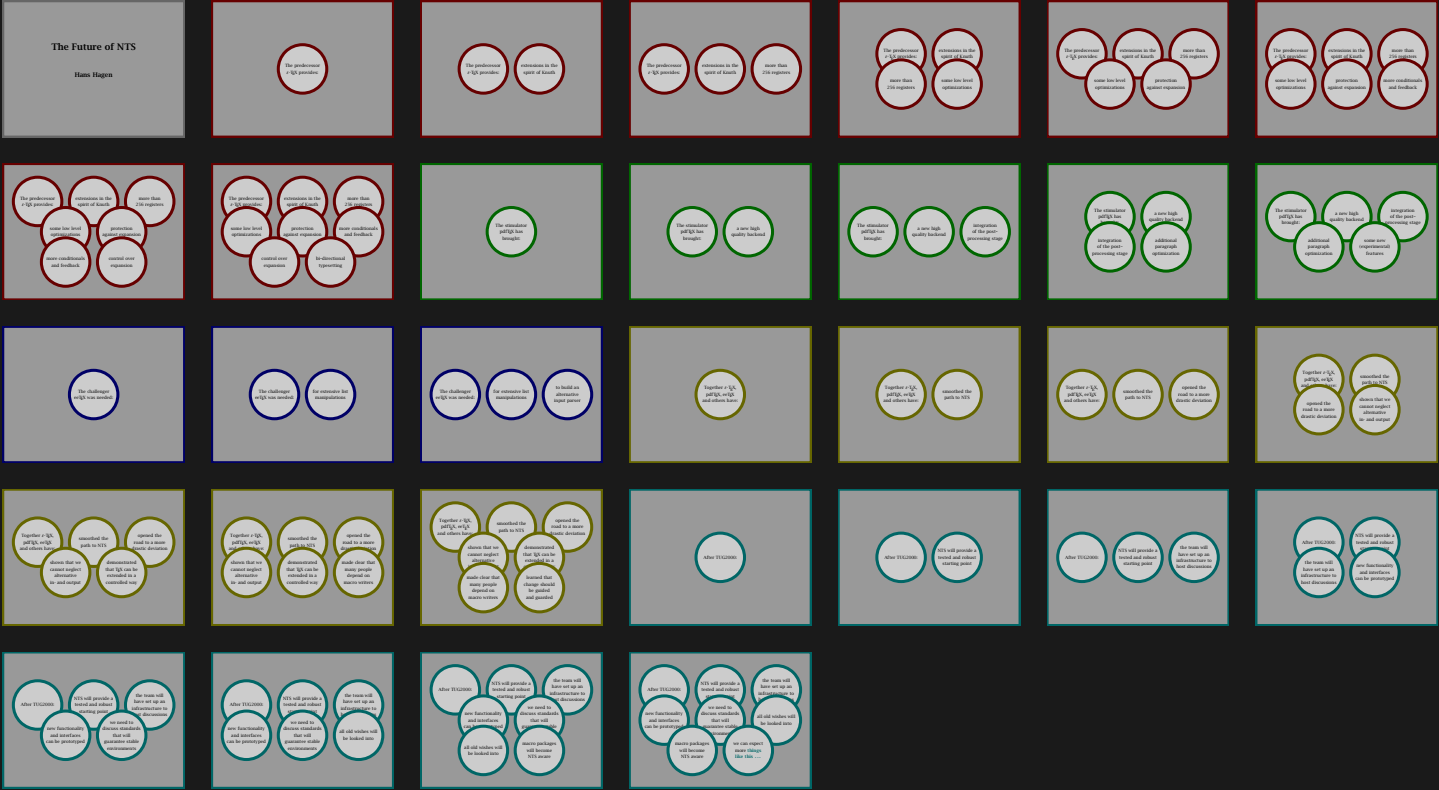
The coding

- UML showed a way but C++ got away with it.
- We can do math (and), so we can do them all.
- It's not only in the coding, but also in the methods.
- We often do we want to rule the job.
- We will probably soon learn what we did forget.
- We had better learn to look into the future.

Conclusions

- Nothing that real surprise is happening.
- We are already 15 years ahead but often unaware of it.
- Coding is not the main issue, the (change in) content is.
- There will be documents, but we will not recognize them as such.
- We have the tools and our tools are coming.
- We have stability and finally people will use it.
- TeX is one of the few life-long tools around.
- For a long time, TeX will be around but nobody will notice.
- There are some millionaires to come, so we can take some time.
- We must not forget the past and not ignore the future.

writing / pre-13



balls / pre-15

Definitions

- TeX
- MetaPost
- Portable Document Format

TeX

- We use the word TeX to both a programming language and a program. The language can be used to describe a document and its format. Documents are called TeX documents and the files.
- Books are collected in a so called document class. The program interprets the source and produces a format output of the document.
- Although TeX originates at the beginning of the eighties of the
- it still one of the most powerful tools available for preparing complex documents.
- Example is public due to the fact that TeX can still read a source

TeX

We use the word TeX to both a programming language and a program. The language can be used to describe a document and its format. Books are collected in a so called document class. The program interprets the source and produces a format output of the document.

Although TeX originates at the beginning of the eighties of the

it still one of the most powerful tools available for preparing complex documents. It is possible to generate a public due to the fact that TeX can still read a source

MetaPost

This program is derived from MetaPost. Like TeX, both interpret a language and a program. Where MetaPost generates images, TeX generates text. MetaPost is using a two column

The graphic capabilities of TeX programs are limited. The latest version generates two level drawings. On the other hand, MetaPost has powerful capabilities. This means that it can use the program for other graphics that have a similar look or feel.

Portable Document Format

Professional printers and other image processors often have a hard time to display the proper page description language. In order to PDF, the portable document format, the document capabilities, the ability to other standard way of describing a document.

The standard source, image, text, table of contents, and support for document the hyperlinks and comments. The self-contained format is used, combined for describing documents across hardware architectures and software platforms.

JavaScript

Also known as the ECMAScript scripting language, JavaScript is an object oriented, interpreted programming language. It is primarily meant to be embedded in documents.

Although it looks similar to Java, JavaScript is powerful enough to deal with user input as well as providing additional programming functionality. JavaScripts JavaScript looks rather basic because for communicating with the remote world, but the source

eXtensible Markup Language

Even when computers started being much people have been trying to describe their documents, and it is not that computers can deal with their structure and content. Creating an XML, Extensible Markup Language (XML), and its associated

These languages can be recognized by the language. The language is used to describe the structure and content of the document. The language is used to describe the structure and content of the document. The language is used to describe the structure and content of the document.

Cascading

A collection of general purpose source written in the TeX macro language is referred to as a source package. When using Cascading, it is used a markup. Cascading allows a wide range of features and can be used for representing paper and visual documents.

MetaPost supports a specific structured and flexible, providing the most advanced graphic capabilities. It is possible to generate a public due to the fact that TeX can still read a source

Educational Typesetting

Hans Hagen, GUT 2001

[Playground](#)
[Conditions](#)
[Observations](#)
[Solutions](#)
[Context](#)

[Home](#)

Playground

- authors want control over their text
- designers want to make fancy things
- publishers want to (re)publish content

[Playground](#)
[Conditions](#)
[Observations](#)
[Solutions](#)
[Context](#)

[Home](#)

Conditions

- materials should follow changes in educational programs
- content must be available in many levels and depths
- information to be reused for formal and job related education
- content should be coded in one source or come from databases
- consistency should be guaranteed within authors teams
- workflows should be automated as much as possible

[Playground](#)
[Conditions](#)
[Observations](#)
[Solutions](#)
[Context](#)

[Home](#)

Observations

- 100 years ago demands were not that high, and it was not that hard to find good **processors (publishing & books)**. T₂E permits you to reach a very high level of abstraction
- today **T₂E** is the **hottest**, so whatever you produce, it should become **hot**, some day. since there are some limitations, in a way it's too become **much more robust**
- **specific** standards pop up faster than one can support them. fighting them will become more and more important
- the publishing world is about to change fast: publishers become brokers of information (**editors**)

[Playground](#)
[Conditions](#)
[Observations](#)
[Solutions](#)
[Context](#)

[Home](#)

Solutions

- accessible input (**input**)
- structured content (**processors & exanams**)
- high quality output (**editors & books**)
- controlling the process (**editors & books**)
- typesetting on demand (**and [progress.pdf.com](#)**)

[Playground](#)
[Conditions](#)
[Observations](#)
[Solutions](#)
[Context](#)

[Home](#)

Context

- general purpose means package is currently in major upgrade stage
- extensive support for structuring and more (in T₂E and T₂M)
- reusable blocks and extensible (more interfaces will be provided)
- evolving into a full-sized toolkit (scripts, data, styles)
- will soon provide DTP like control and features

[Playground](#)
[Conditions](#)
[Observations](#)
[Solutions](#)
[Context](#)

[Home](#)

Postprocessing PDF

What is PDF

For long time PDF was X³PS's native output format. This format can be converted to for instance POSTSCRIPT or PDI. The later format has the advantage that fonts and graphics are embedded which make the file portable across platforms. We start this day with a short explanation of what PDF is.

Hans Hagen

Postprocessing PDF

The history of PDF¹X

The PDF¹X project started ... years ago. In its current incarnation, this program is rather stable and mature. However, it took quite some development, discussion and testing, and the PDF¹X mailing list has played an important role in this. In this regard, this project can be considered one of the most innovative X³ related activities of the end of the previous century. How did it all evolve?

Sebastian Rahtz

Postprocessing PDF

Fonts in PDF¹X

Since PDF¹X provides its own backend, it also has to deal with font inclusion. PDF¹X supports type 1 as well as true-type and bitmap fonts. Some can be included directly, others needs special treatment. Fonts can be embedded completely, partially, or not at all. Also, users have to set up some map files. Although font support is rather straightforward, some basic knowledge can be handy.

Erik Frahmach

Postprocessing PDF

How PDF¹X can improve your pages

It may have gone unnoticed to many happy users, but one of the main reasons for developing PDF¹X was the wish to improve the visual appearance of the page. The current nature of X³ The Program, limits this improvement to the individual paragraphs and pages. Currently PDF¹X provides several methods to improve the look and feel of a page. Systematic experiments and research were the basis for the evolution of PDF¹X.

Han Thô Thinh

Postprocessing PDF

Graphics in PDF¹X

A consequence of being its own backend, is that PDF¹X must include graphics itself. PDF¹X supports the PDI, J²C, P³NG and vector⁴ graphic formats. PDI graphics can be converted to PDI. Because PDF¹X gives you access to low level PDI, it can also support that resolution graphics. When embedding graphics one has to consider resolution and color.

Hans Hagen

Postprocessing PDF

PDF¹X in a workflow

Since PDF is one of the major file formats, PDF¹X is a good candidate for acting as a backend in processing data. How does that work, and what is needed to get it working.

Ed Cashin

Postprocessing PDF

Going beyond static documents

The last few years, the world of documents has changed drastically. Color has become natural on the desktop and e-reports document go beyond their static counterparts. One way to enhance documents is to use advanced hypertext tricks. A more drastic deviation from traditional documents is embedding program code, like J⁵AVASCRIPT. One can use this scripting language to provide controllable navigation and intelligence to documents. PDF¹X provided the hooks to embed such scripts into the document. In a similar way, one can use PDF¹X to make advanced forms.

Hans Hagen

Postprocessing PDF

Setting up PDF¹X

Since PDF¹X is an all-in-one tool, the X³ user no longer has to deal with a multi-stage source to paper process. Installation is not that complicated, but there a few thing you should know about the configuration.

Ed Cashin

Postprocessing PDF

Postprocessing PDF

It is not uncommon to postprocess the files produced by X³, for instance making X³ headers out of A⁶d documents. Since PDF¹X can process PDF graphics, it can do its own advanced postprocessing, sometimes going far beyond what's common in the X³ world. Another kind of postprocessing involves converting PDF into a textual format. An example of this application is an experimental utility that converts X³ into HTML, in a rather natural way.

Berend de Boer

cycle / pre-22

Something
Very Important

August 2000

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Alpha

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Beta and Gamma

A simple and not too long text just to show the idea.

Delta

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Epsilon

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Zeta, Eta and Theta

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Omega


```
module pre-01
colors BackgroundColor InteractionColor ContrastColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
\Subject{}
```

```
module pre-02
colors BackgroundColor OrnamentColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
\Subject{}
```

```
module pre-03
colors PageColor BackgroundColor ContrastColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
module pre-04
colors red green blue yellow gray
```

```
\TitlePage{}
\Topics{}
\Topic{}
```

```
module pre-05
colors BackgroundColor OrnamentColor

\TitlePage{}
\Topic{}
```

```
module pre-06
colors PageColor BackgroundColor ContrastColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
module pre-07
colors PageColor TextColor LineColor linecolor 1-6
```

```
\TitlePage{}
\Topics{}
\StartIdeas
  \Topic{}
  \StartIdea
  \StopIdea
\StopIdeas
```

```
module pre-08
colors PageColor TextColor LineColor [ColorPage ColorLine]
layers text sample idea
modes demo
```

```
\TitlePage{}
\StartIdea
  \StartSample \StopSample
  \StartText   \StopText
\StopIdeas
```

```
module pre-09
colors PageColor TextColor LineColor InteractionColor
modes demo
```

```
\TitlePage{}
\Topic{}
\StartIdea
  \Topic{}
  \StartSample \StopSample
  \StartText \StopText
\StopIdeas
\StartText
\StopText
```

```
module pre-10
colors PageColor TextColor LineColor InteractionColor
modes demo
```

```
\TitlePage{}
\Topic{}
\StartIdea
  \Topic{}
  \StartSample \StopSample
  \StartText \StopText
  \StartSubText \StopSubText
\StopIdeas
\StartSample
\StopSample
\StartText
\StopText
\StartSubText
\StopSubText
```

```
module pre-13
colors    TextColor PageColor LineColor SymbolColor

\TitlePage{}
\Topics{}
\Topic{}
```

module pre-14
colors One Two

\TitlePage{}

\Topic{}

\Subject{}

```
module pre-15
colors  TextColor PageColor LineColor linecolor 1-6
modes  demo
```

```
\TitlePage{}
\StartIdea
  \StartItem
  \StopItem
\StopIdea
```

```
module pre-16
colors  TextColor PageColor LineColor
modes  demo
```

```
\TitlePage{}
\StartIdea
  \Topic{}
  \NextIdea
\StopIdea
```

```
module pre-19
colors  TextColor PageColor OrnamentColor InteractionColor ContrastColor
modes  demo
```

```
\TitlePage{}
```

```
\Topic{}
```

```
module pre-22
colors TopColor BotColor DotColor
modes demo
```

```
\TitlePage{}{}
\StartSummary{}{}
\StopSummary
```

```
module pre-23
colors TopColor BotColor DotColor
modes demo
```

```
\TitlePage{}{}
\StartSummary{}{}
\StopSummary
```

```
module pre-26
modes demo reverse
```

```
\TitlePage{}
\StartTopic
\Title{}
\StopTopic
\ColofonPage{}
```