

Introduction à ConTEXt

Denis Roegel

Journée GUTenberg

29 mai 2002

Introduction

- On m'a demandé de faire une introduction à ConT_EXt.
- Qu'est-ce qu'une introduction à ConT_EXt ?
- Ou, en d'autres termes, de quoi suis-je susceptible de parler aujourd'hui ?

fermer

Il y a plusieurs possibilités :

1. Je peux vous présenter complètement ConTEXt, chaque commande, et toutes ses possibilités ;
2. je peux vous donner des aperçus de choses faites avec ConTEXt, ou faire un petit panorama ;
3. je peux vous donner quelques éléments de ConTEXt, quelques *repères* pour vous donner confiance ;

fermer

4. je peux faire une comparaison avec L^AT_EX ou d'autres formats ;

5. etc.

fermer

Objectif de cette présentation

- Si l'on m'a demandé de faire cette présentation, c'est je pense pour vous permettre de tester ConTEXt, pour vous donner les *moyens* de passer à ConTEXt.
- Beaucoup d'entre vous ont déjà eu un aperçu de ConTEXt, au travers des présentations de Hans Hagen. Mon objectif principal n'est donc pas de vous convaincre que c'est un système intéressant, mais de vous montrer *comment vous pouvez vous y mettre*.

fermer

- L'exposé va donc être un ensemble de repères destinés à vous mettre en confiance et à vous donner envie et les moyens d'aller voir de plus près.
- Ce ne sera pas un cours de ConTExT, ni même une introduction au sens habituel du terme. Il faudra compléter ce document par bien d'autres documents, mais ce document devrait permettre de débroussailler un peu le terrain.

fermer

Historique

- ConT_EXt a été développé à partir de 1990 essentiellement par Hans Hagen. D'autres personnes sont impliquées de nos jours, notamment Taco Hoekwater.
- ConT_EXt a été développé à des fins commerciales et c'est le gagne-pain de Hans Hagen. Ceci en assure un développement constant.
- Moyennant quelques restrictions, ConT_EXt est utilisable par n'importe qui.

fermer

Qu'est-ce que ConT_EXt ?

- ConT_EXt repose en grande partie sur plain T_EX et n'a pratiquement pas de relations avec L^AT_EX.
- ConT_EXt est un *système intégré d'outils typographiques* basés sur T_EX, pas uniquement un format.

fermer

Quelques avantages de ConT_EXt

- système intégré, permettant une bonne coopération de différents outils, notamment T_EX et METAPOST ;
- interface simple, homogène et extensible ;
- très bon support pour la génération de PDF et exploitation des fonctionnalités de pdfT_EX ;
- possibilité de créer des documents interactifs ;
- système pratiquement monolithique, assurant une bonne intégration des différents compo-

fermer

sants ; presque tout ce qui est utile est intégré dans le noyau.

fermer

Comparaison avec L^AT_EX

- de nombreux *packages*, aux fonctionnalités quelquefois redondantes, et incompatibles entre eux ;
- interfaces non homogènes ; le passage de paramètre par *clé* et *valeur* n'est utilisé que par quelques *packages*, par exemple `graphicx` ;
- certaines commandes sont difficilement extensible (par exemple `\section`) ;
- il faut des *packages* spécialisés pour obtenir des documents interactifs (par exemple `pdfscreen`), mais ceux-ci ne sont pas forcément

fermer

facilement paramétrables, en particulier vis-à-vis des graphiques.



fermer

Exemple

- Un exemple d'intégration entre ConTEXt et METAPOST : ce document.
- En exercice, faites la même chose avec votre outil préféré.

fermer

Plan de la suite

Après cette introduction, je vais diviser mon exposé en plusieurs parties. L'ordre choisi correspondra plus ou moins à l'apprentissage de celui qui installe ConTEXt pour la première fois. Cette présentation est donc volontairement plus proche d'un TD que d'un cours magistral.


1. un fichier ConTEXt élémentaire et sa compilation ;
2. structure générale d'un fichier source ;
3. un exemple plus complexe, avec une inclusion graphique générée à la volée ;

fermer



4. examen de l'installation de ConT_EXt sous T_EXlive ;
5. les fichiers de ConT_EXt ;
6. la documentation de ConT_EXt ;
7. examen détaillé de la flexibilité/configurabilité d'une commande ;
8. le support du français ;
9. la gestion des polices et comparaison avec la gestion de L^AT_EX (i.e. NFSS) ;

fermer



10. retour sur MetaFun, l'intégration de META-
POST dans ConTEXt ;

11. divers autres aspects.

fermer

Un fichier ConTeXt élémentaire

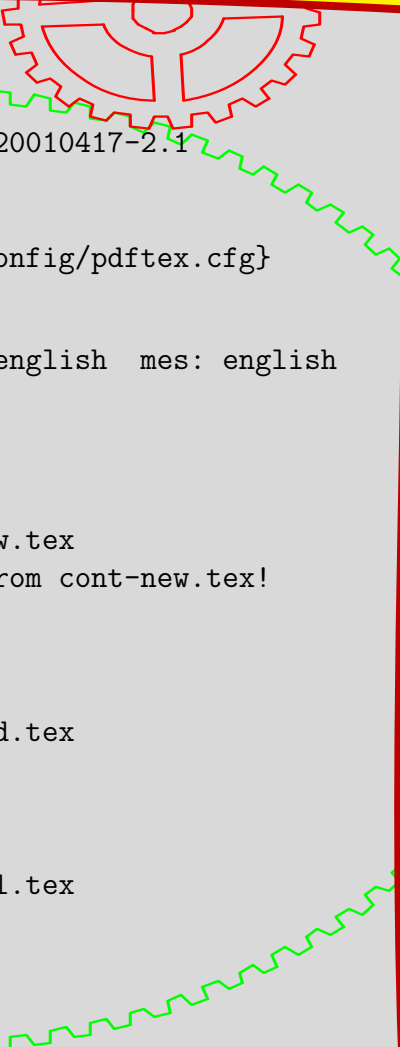
```
\starttext  
hello, world!  
\stoptext
```

Sa compilation :

```
context 211 % texexec test7  
TeXExec 2.8 - ConTeXt / PRAGMA ADE 1997-2002
```

```
executable : pdfetex  
format : cont-en  
inputfile : test7  
output : standard  
interface : en  
current_mode : none  
TeX run : 1
```

fermer




```
This is pdfTeX, Version 3.14159-14h-released-20010417-2.1
(Web2C 7.3.3.1)
entering extended mode
(./test7.tex{/usr/share/TeX/texmf-var/pdftex/config/pdftex.cfg})
```

```
ConTeXt ver: 2002.5.15 fmt: 2002.5.25 int: english mes: english
```

```
language      : language en is active
system        : cont-new loaded
(/usr/share/TeX/texmf/tex/context/base/cont-new.tex
systems       : beware: some patches loaded from cont-new.tex!
system (E-TEX) : [line 839] \dimexpr
)
system        : cont-old loaded
(/usr/share/TeX/texmf/tex/context/base/cont-old.tex
loading       : Context Old Macros
)
system        : cont-fil loaded
(/usr/share/TeX/texmf/tex/context/base/cont-fil.tex
loading       : Context File Synonyms
)
system        : cont-sys loaded
```


fermer



```
(/usr/share/TeX/texmf/tex/context/user/cont-sys.tex)
bodyfont      : 12pt rm is loaded
language      : patterns 2:2-en-2 4:2-de-2 5:2-fr-2 loaded
specials      : tex,postscript,rokicki loaded
system        : test7.top loaded
(./test7.top)
pdftex        : needs map file: original-context-symbol.map
pdftex        : needs map file: pl0-ams-cmr.map
pdftex        : needs map file: original-vogel-symbol.map
systems       : begin file test7 at line 1
[1 1]
systems       : end file test7 at line 3
)
Output written on test7.dvi (1 page, 304 bytes).
Transcript written on test7.log.
```

```
run time : 3 seconds
sorting and checking : running texutil
```

```
TeXUtil 7.3 - ConTeXt / PRAGMA ADE 1992-2000
```



```
action : processing commands, lists and registers
option : sorting IJ under Y
option : converting high ASCII values
input file : test7.tui
output file : test7.tuo
passed commands : 7
  remapped keys : 0
register entries : 0 -> 0 entries 0 references
synonym entries : 0 -> 0 entries
embedded files : 1
```

```
utility file analysis : another run needed
  TeX run : 2
```

```
This is pdfTeX, Version 3.14159-14h-released-20010417-2.1
```

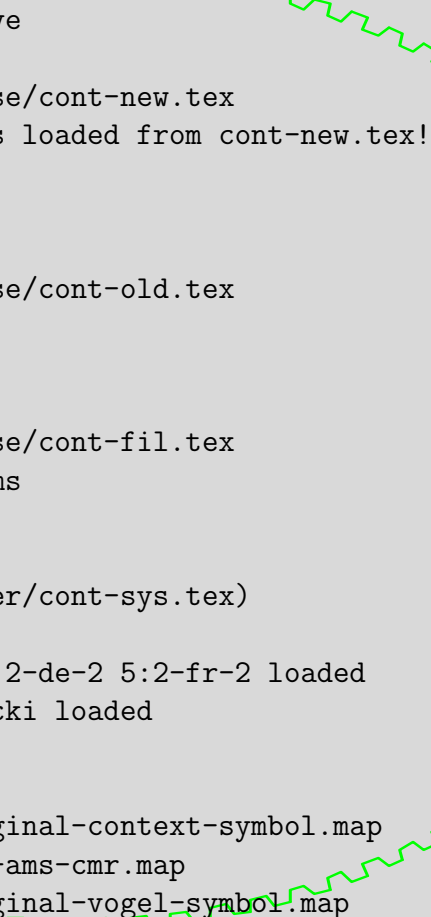
```
(Web2C 7.3.3.1)
```

```
entering extended mode
```

```
(./test7.tex{/usr/share/TeX/texmf-var/pdftex/config/pdftex.cfg}
```


```
ConTeXt ver: 2002.5.15  fmt: 2002.5.25  int: english  mes: english
```

fermer



```
language      : language en is active
system       : cont-new loaded
(/usr/share/TeX/texmf/tex/context/base/cont-new.tex
systems      : beware: some patches loaded from cont-new.tex!
system (E-TEX) : [line 839] \dimexpr
)
system       : cont-old loaded
(/usr/share/TeX/texmf/tex/context/base/cont-old.tex
loading      : Context Old Macros
)
system       : cont-fil loaded
(/usr/share/TeX/texmf/tex/context/base/cont-fil.tex
loading      : Context File Synonyms
)
system       : cont-sys loaded
(/usr/share/TeX/texmf/tex/context/user/cont-sys.tex)
bodyfont     : 12pt rm is loaded
language     : patterns 2:2-en-2 4:2-de-2 5:2-fr-2 loaded
specials     : tex,postscript,rokicki loaded
system       : test7.top loaded
(./test7.top)
pdftex      : needs map file: original-context-symbol.map
pdftex      : needs map file: pl0-ams-cmr.map
pdftex      : needs map file: original-vogel-symbol.map
```

fermer




```
(./test7.tuo) (./test7.tuo) (./test7.tuo)
(./test7.tuo) (./test7.tuo)
(./test7.tuo) (./test7.tuo) (./test7.tuo)
(./test7.tuo) (./test7.tuo)
(./test7.tuo) (./test7.tuo) (./test7.tuo)
systems      : begin file test7 at line 1
[1.1]
systems      : end file test7 at line 3
)
Output written on test7.dvi (1 page, 304 bytes).
Transcript written on test7.log.
```

```
run time : 1 seconds
sorting and checking : running texutil
```

TeXUtil 7.3 - ConTeXt / PRAGMA ADE 1992-2000

```
action : processing commands, lists and registers
option : sorting IJ under Y
option : converting high ASCII values
input file : test7.tui
output file : test7.tuo
```




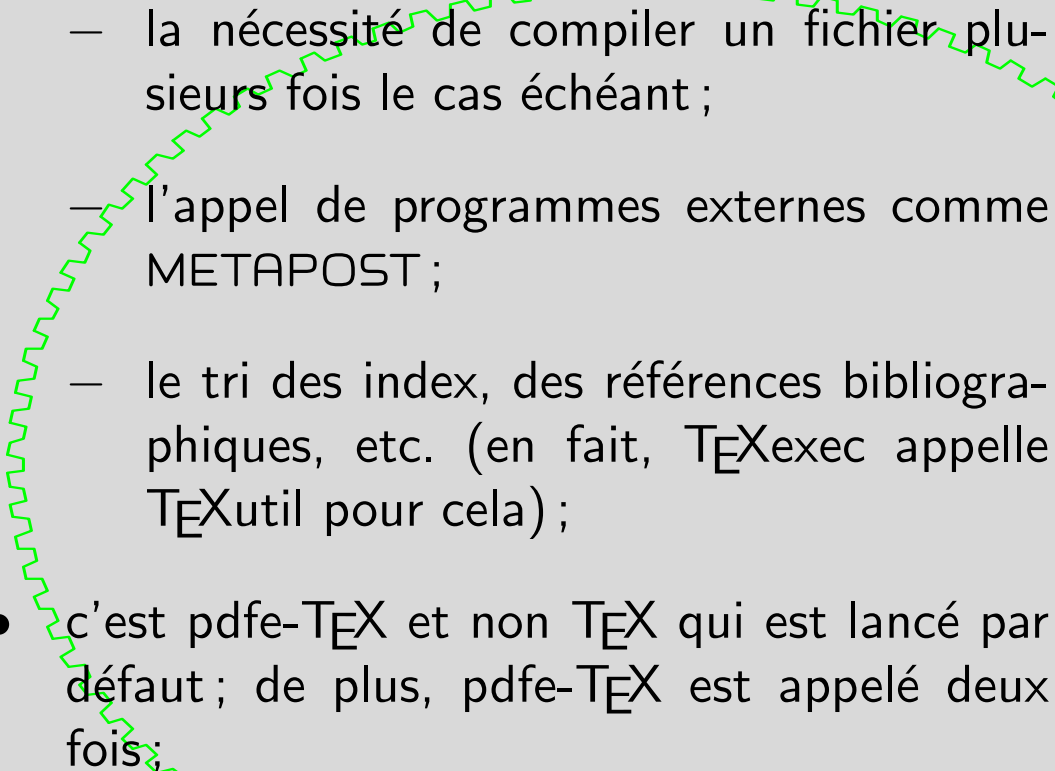
```
passed commands : 7
  remapped keys : 0
register entries : 0 -> 0 entries 0 references
synonym entries : 0 -> 0 entries
  embedded files : 1

total run time : 5 seconds
context 212 %
```


Observations :

- nous avons lancé non pas $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, ni même directement $\text{T}_{\text{E}}\text{X}$, mais un script $\text{T}_{\text{E}}\text{X}_{\text{exec}}$, écrit en Perl et qui se charge de la plupart des détails de compilation, notamment :

fermer

- 
- 
- la nécessité de compiler un fichier plusieurs fois le cas échéant ;
 - l'appel de programmes externes comme METAPOST ;
 - le tri des index, des références bibliographiques, etc. (en fait, $\text{T}_{\text{E}}\text{X}_{\text{exec}}$ appelle $\text{T}_{\text{E}}\text{X}_{\text{util}}$ pour cela) ;
 - c'est pdf $\text{T}_{\text{E}}\text{X}$ et non $\text{T}_{\text{E}}\text{X}$ qui est lancé par défaut ; de plus, pdf $\text{T}_{\text{E}}\text{X}$ est appelé deux fois ;

fermer

- 
- le résultat produit est un fichier Dvi ; un fichier PDF peut être obtenu en ajoutant l'option

`--pdf`

à l'appel de $\text{T}_{\text{E}}\text{X}_{\text{exec}}$; il est aussi possible d'ajouter un commentaire spécial au début du fichier source pour obtenir le même résultat.

fermer



Structure générale d'un fichier source

Un certain nombre de concepts sont présents aussi bien dans ConTEXt que dans L^ATEX :

- notion de préambule :
ce qui précède `\starttext` ;
- le corps du document (environnement `document` en L^ATEX) :
ce qui est entre `\starttext` et `\stoptext` ;
- le préambule est le lieu des configurations globales, ces configurations pouvant être localement modifiées ;

fermer



- différentes commandes parsèment le texte, ces commandes étant comme en L^AT_EX ou bien du type `\commande{paramètres}` (ou des variantes que l'on verra), ou bien des environnements faits de deux commandes, l'une `\start...`, l'autre `\stop...` ;
- ConT_EXt s'appuie sur plain T_EX, et la plupart des commandes de plain T_EX sont donc disponibles ; on peut en particulier définir des commandes avec `\def`.

fermer



Environnements


- Beaucoup d'environnements ont des équivalents, plus ou moins immédiats. Par exemple,

```
\begin{itemize}  
\item ...  
\end{itemize}
```

deviendra :

```
\startitemize  
\item ...  
\stopitemize
```

fermer




Dans beaucoup de cas, l'adaptation d'un fichier L^AT_EX ne sera pas beaucoup plus complexe que cela.

- Certains environnements de ConT_EXt regroupent — grâce à des options — plusieurs environnements de L^AT_EX. Par exemple, l'équivalent de l'environnement `enumerate` est obtenu ainsi :

```
\startitemize[n]  
\item ...  
\stopitemize
```

fermer



La commande `\startitemize` reçoit ici un paramètre correspondant à une liste numérotée.

- Parmi les environnements courants, celui qui demande le plus d'adaptation est probablement celui permettant de réaliser des tableaux. ConTEXt s'appuie sur les macros TaBLE développées par Michael Wichura et ces macros sont proches du codage d'un tableau avec la primitive `\halign`. En particulier, les lignes verticales doivent être positionnées explicitement.

fermer



```
\starttable[|l|c|r|
```

```
\HL
```

```
\VL aaaa \VL b \VL c\VL\FR
```

```
\HL
```

```
\VL d \NC eeee \NC ffff\VL\MR
```

```
\HL
```


```
\VL g \VL h \VL i\VL\LR
```

```
\HL
```

```
\stoptable
```

aaaa	b	c
d	eeee	ffff
g	h	i

fermer



- Environnement de numérotation de lignes :

```
\startlinenumbering
```

Voici quelques lignes de texte, qui dans l'environnement `\type{linenumbering}` sont automatiquement numérotées.

Voici quelques lignes de texte, qui dans l'environnement `\type{linenumbering}` sont automatiquement numérotées.

Voici quelques lignes de texte, qui dans l'environnement `\type{linenumbering}` sont automatiquement numérotées.

```
\stoplinenumbering
```

- 1 Voici quelques lignes de texte, qui dans l'environnement `linenumbering`
- 2 sont automatiquement numérotées. Voici quelques lignes de texte, qui dans
- 3 l'environnement `linenumbering` sont automatiquement numérotées. Voi-
- 4 ci quelques lignes de texte, qui dans l'environnement `linenumbering` sont
- 5 automatiquement numérotées.

- etc.

fermer

Exemple avec graphique automatique

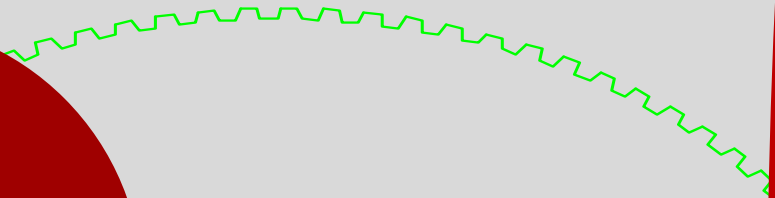
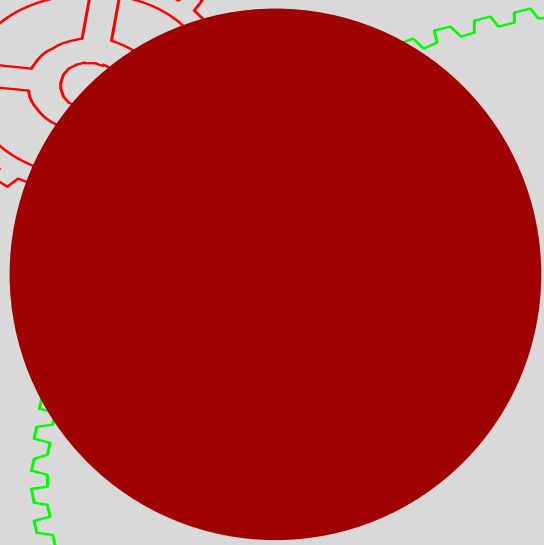
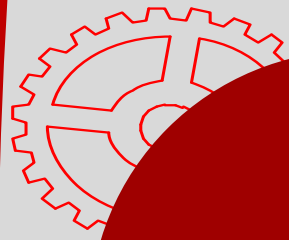
Le fichier suivant montre comment un dessin METAPOST peut être défini au sein d'un fichier ConT_EXt et comment ce dessin peut être inclus sans que l'on n'ait jamais à se préoccuper de la compilation d'un fichier METAPOST.

```
\startMPgraphic  
fill fullcircle scaled 200pt withcolor .625red ;  
\stopMPgraphic
```

```
\loadcurrentMPgraphic
```


```
\placeMPgraphic
```

fermer



- Les dessins créés peuvent avoir des noms et un dessin peut en inclure un autre, ce qui permet de les construire incrémentalement.
- Il est possible de distinguer entre des dessins uniques, régénérés à chaque fois, et des des-

fermer



sins qui ne changent pas d'une utilisation à l'autre.

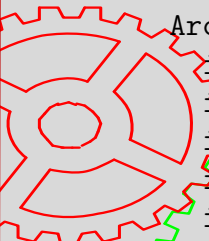
- **Avantage de l'inclusion :** le dessin peut être facilement rendu dépendant de son environnement. C'est par exemple ce qui se passe avec le fond de mes transparents. Ils ne sont pas dans un fichier METAPOST, mais intégrés dans le source du document, et ils peuvent ainsi facilement dépendre du numéro de page.

fermer

Installation sous T_EXlive

- ConT_EXt s'installe assez simplement.
- Comme ConT_EXt évolue beaucoup, la version sur T_EXlive doit en général être mise à jour. La version sur T_EXlive 7 comporte ainsi quelques bugs qui ont déjà été corrigés.
- On cherchera donc sur www.pragma-ade.com la dernière version de ConT_EXt. Il s'agit en principe d'un fichier `cont-tmf.zip`.
- Pour installer cette archive, il faut la placer à la racine d'une arborescence TDS et lancer `unzip -a :`

fermer



```
test 225 % unzip -a cont-tmf.zip
```

```
Archive:  cont-tmf.zip
```

```
  inflating: context/config/texexec.rme
```

```
  inflating: context/perlTk/runperl.zip
```

```
  inflating: context/perlTk/texexec.rme
```

```
  inflating: tex/context/base/context.tex
```

```
  inflating: tex/context/base/cont-cz.tex
```

```
  inflating: tex/context/base/cont-de.tex
```

```
...
```

```
  inflating: metapost/context/mp-form.mp
```

```
  inflating: metapost/context/mp-symb.mp
```

```
  inflating: metapost/context/metafun.mp
```

```
  inflating: doc/context/base/mreadme.pdf
```

```
  inflating: doc/context/base/minstall.pdf
```

```
  inflating: fonts/tfm/hoekwater/context/contnav.tfm
```

```
  inflating: fonts/type1/hoekwater/context/contnav.pfb
```

```
  inflating: dvips/config/context.map
```

```
  inflating: pdftex/config/context/original-youngryu-px.map
```

```
  inflating: pdftex/config/context/original-youngryu-tx.map
```

```
  inflating: pdftex/config/context/il2-ams-cmr.map
```

```
  inflating: pdftex/config/context/pl0-ams-cmr.map
```


```
  inflating: pdftex/config/context/original-ams-cmr.map
```

```
  inflating: pdftex/config/context/original-ams-euler.map
```

```
  inflating: pdftex/config/context/original-adobe-euro.map
```

fermer

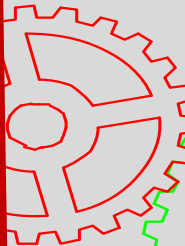
```
inflating: pdftex/config/context/original-context-symbol.map  
inflating: pdftex/config/context/original-vogel-symbol.map
```



Les fichiers sont donc automatiquement mis au bon endroit.

- L'étape suivant consiste à mettre au bon endroit les fichiers Perl. Ceux-ci ont été mis sous `context/perltk` et doivent être copiés en un endroit où les exécutable sont cherchés. Sur mon système, je les ai mis avec les autres exécutable de $\text{T}_{\text{E}}\text{X}$. Il faut de plus faire quelques liens comme `texexec` qui doit pointer vers `texexec.pl`. Tous ces points

fermer



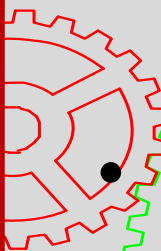
sont détaillés dans un fichier de documentation de ConTEXt appelé `mininstall.pdf` et donc la lecture est recommandée à l'installation. C'est un document très court, mais indispensable.

- Ce même document explique comment créer un format, le plus simple étant de taper :

```
texexec --make en
```

- Il est possible qu'ensuite le format doive être mis manuellement à la bonne place, ceci dépendant des privilèges sous lesquels la commande `texexec` aura été appelée.

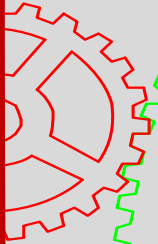
fermer

- 
- Le système contient aussi un fichier `cont-sys.rme` qui doit être renommé en `cont-sys.tex` et permet de préciser certains comportements par défaut du système.
 - Une fois ces fichiers installés, il ne reste qu'à régénérer la base de donnée du système avec par exemple `texhash` et une commande similaire pour faire connaître à l'environnement les nouveaux exécutable. `ConTEXt` devrait alors être opérationnel.

L'ensemble de ces opérations peut être fait en quelques cinq minutes.

fermer

Il est une fois de plus conseillé de lire le fichier
minstall.pdf.



fermer

Les fichiers de ConT_EXt

Après l'installation, il est instructif de jeter un coup d'œil aux fichiers de l'installation.

- Le répertoire principal est `tex/context/base` qui contient 333 fichiers. La taille globale de ces fichiers est d'environ 5 Mo.
- L'un des fichiers est le fichier `context.tex` qui comporte plus d'une centaine de commandes `\input` chargeant les autres parties du système à la création du format.
- À titre de comparaison, le noyau de L^AT_EX comporte environ 126 fichiers totalisant 1 Mo.

fermer

Mais ConTEXt inclut de nombreuses fonctionnalités qui ne se trouvent pas dans le noyau de L^ATEX, mais dans divers *packages*, ce qui rend la comparaison un peu difficile.

- Il n'en reste pas moins que le format de ConTEXt est particulièrement gros :

```
4029973 May 25 11:19 cont-en.efmt
750724 May 19 11:25 latex.fmt
382464 May 21 17:02 metafun.mem
```

- Le format ConTEXt est aussi nettement plus gourmand en ressources TEX, mais les distributions courantes sont configurées pour en tenir compte. Dans certains cas, cependant,

fermer

il peut être nécessaire de modifier l'environnement en adaptant les valeurs des variables spécifiées dans `texmf.cnf`.



fermer

Jetons un coup d'œil dans un fichier pris au hasard, par exemple `page-mar.tex` :

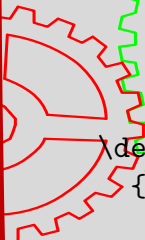
```
%D \module
%D [      file=page-mar, % moved here from main-001
%D      version=1997.03.31,
%D      title=\CONTEXT\ Core Macros,
%D      subtitle=Marginal Things
%D      author=Hans Hagen,
%D      date=\currentdate,
%D      copyright={PRAGMA / Hans Hagen \& Ton Otten}]
%C
%C This module is part of the \CONTEXT\ macro||package and is
%C therefore copyrighted by \PRAGMA. See mreadme.pdf for
%C details.

%D This module is still to be split and documented.

\writestatus{loading}{Context Core Macros / Marginal Things}

\unprotect
```

fermer




```
\newif\iflowinmargin

\def\stelinmargein
  {\dodoubleempty\dostelinmargein}

\def\dostelinmargein[#1][#2]%
  {\ifsecondargument
   \processcommalist[#1]{\dodostelinmargein[#2]}% becomes [#2]{##1}
  \else
   \getparameters[\??im][#1]%
  \fi}

\def\dodostelinmargein[#1]#2% [settings]{class}
  {\ifundefined{\??im#2\c!offset}%
   \presetlocalframed
     [\??im#2]%
   \getparameters
     [\??im#2]
     [\c!kader=\v!uit,
      \c!offset=\v!overlay,
      \c!regel=1,
      \c!scheider=,
```

fermer



```
\c!breedte=\v!ruim,  
\c!afstand=\!!zeropoint,  
\c!letter=\@@imletter,  
\c!kleur=\@@imkleur,  
\c!plaats=\@@implaats,  
\c!uitlijnen=\@@imuitlijnen,  
\c!voor=\@@imvoor,  
\c!na=\@@imna,  
#1]%  
\else  
  \getparameters[\??im#2][#1]%  
\fi}  
  
\let\margetekstafstand = \!!zeropoint  
\def\margetekstregels {1}  
\def\margetekstnummer {0}  
\let\margetekstscheider = \empty  
...  
\stelnaastplaatsenin  
  [\c!status=\v!stop]  
  
\protect \endinput
```

fermer

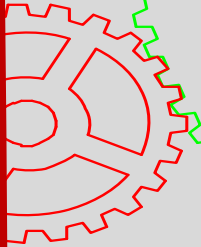
Observations :

- le code ressemble à du $\text{T}_{\text{E}}\text{X}$, ce qui est rassurant ;
- de manière interne, plusieurs caractères sont rendus équivalents à une lettre, notamment le ! ; le passage du mode protégé au mode non protégé se fait avec `\unprotect` (analogue de `\makeatletter` et le retour de ce mode avec `\protect` (analogue de `\makeatother`) ; ces commandes peuvent s'utiliser dans le corps du document ;
- le code source comporte de manière interne des mots en néerlandais, vestige d'une époque



fermer

où ConTEXT n'était sans doute pas encore beaucoup utilisé en dehors de PRAGMA ; il est à supposer que ces commandes internes seront peu à peu traduites en anglais.



fermer

La documentation

Le site de PRAGMA contient de nombreux documents, quelquefois enfouis. Les principaux sont les suivants :

- `art-pres.pdf` : *TEX as presentation tool* (8 pages sur les styles prédéfinis pour des présentations interactives) ;
- `cont-eni.pdf` (version interactive) ou `cont-emp.pdf` (version papier) : manuel de ConTeXt (369 pages en version interactive) ;



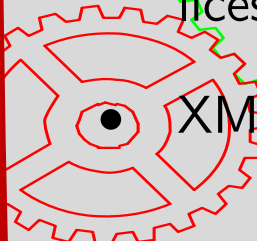
fermer

- `mchart.pdf` : module d'organigrammes (20 pages)
- `metafun-p.pdf` (version papier), `metafun-s.pdf` (version écran) : manuel de MetaFun (364 pages papier, 635 pages écran)
- `mfonts.pdf` : les polices dans ConTEXt (39 pages)
- `minstall.pdf` : l'installation de ConTEXt (13 pages)
- `mp-cb-en.pdf` : ConTEXt, an excursion (129 pages)



fermer

- `mtexexec.pdf` : manuel de $\text{T}_{\text{E}}\text{X}_{\text{exec}}$ (11 pages)
- `mtexfont.pdf` : manuel de $\text{T}_{\text{E}}\text{X}_{\text{font}}$ (17 pages)
- `mtexutil.pdf` : manuel de $\text{T}_{\text{E}}\text{X}_{\text{util}}$ (7 pages)
- `mwidget-p.pdf` (version papier), `mwidget-s.pdf` (version écran) : manuel des Widgets (46 pages écran)
- `showfont.pdf` : exemples d'utilisation de polices (16 pages)

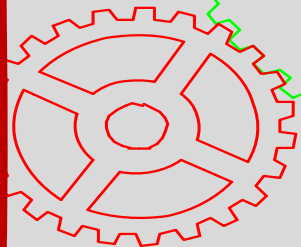


• XML in Con $\text{T}_{\text{E}}\text{X}_{\text{t}}$

fermer

- manuel de `makempy` pour créer des polices outline
- la thèse de Thanh
- etc.


Pour ne pas être noyé dans cette documentation, il faut procéder *méthodiquement*. Je donnerai en fin d'exposé ma méthodologie pour un utilisateur de `LATEX` qui souhaite se lancer dans `ConTEXt`.



fermer

Syntaxe détaillée d'une commande

Chaque environnement peut être configuré grâce à une commande `\setup...` associée. Par exemple, la mise en page peut être contrôlée avec la commande `\setuplayout`. Son argument, entre crochets, est une liste de couples clé = valeur. Quelques unes des clés sont :

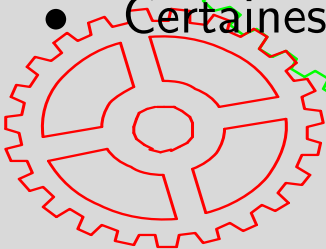


```
width  
height  
backspace  
topspace  
margin  
leftmargin  
rightmargin  
header  
footer  
top
```

fermer

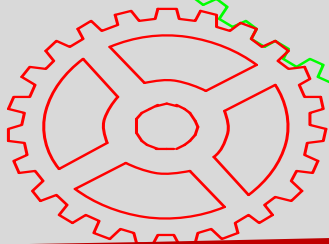
bottom
leftedge
rightedge
headerdistance
footerdistance
topdistance
bottomdistance
leftmargindistance
rightmargindistance
leftedgedistance
rightedgedistance
grid
...

- Les clés correspondent ici aux différents paramètres définissant la mise en page.
- Certaines des clés ont une valeur par défaut.



fermer

- La valeur de certaines clés peut être une commande, ce qui permet de définir des comportements non prévus par défaut.
- Toute la syntaxe est explicitée dans l'excellent manuel (cf. p. 32).



fermer

Support du français

Un début de support pour le français a été mis en place il y a quelques deux semaines ! Auparavant, la configuration pour le français était insuffisante. Même si les choses ne sont pas encore parfaites, elles vont rapidement s'améliorer et les Français peuvent maintenant sans crainte se lancer dans ConT_EXt.

- Tout comme dans L^AT_EX, il faut préciser le codage des caractères en entrée. Ce codage en entrée est appelé régime et est activé par `\enableregime`. Pour le codage ISO-Latin-1, la commande sera :



fermer

```
\enableregime[i11]
```

- Quelques définitions propres au français sont activées par

```
\useencoding[ffr]
```

Ces définitions sont pour l'instant sommaires, mais seront rapidement complétées, en concertation avec les responsables français.

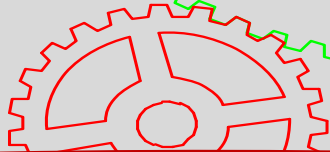
- Enfin, le choix de la langue, en particulier pour les coupures de mots, se fait avec :

```
\mainlanguage[fr]
```



fermer

`\mainlanguage` correspond à la langue principale du document (celle des mots clés, etc.), mais la langue peut être changée localement avec `\language`.



fermer

Index

- Tout comme avec L^AT_EX, un mot est indexé avec la commande `\index`.
- L'index est placé dans le document avec `\placeindex`.
- C'est T_EXexec qui fait le tri et l'utilisateur n'a pas besoin de lancer un programme extérieur comme `makeindex`.



fermer

- Des sous-entrées sont créées avec « + » :
`\index{a+b}`.
- La classification peut être modifiée avec une clé mise entre crochets : `\index[a+b]{\a+\b}`.



fermer

Bibliographie

ConTEXt est muni d'un module `bib` qui permet de gérer une bibliographie. Le format standard de représentation d'une entrée bibliographique est un format `TEX`, et non `BibTEX`

```
\startpublication[k=Brodie84,  
  t=article,  
  a=L.~Brodie,  
  y=1984,  
  s=LB84]  
\artauthor[] {Leo} [L.] {} {Brodie}  
\arttitle{Thinking Forth, a language and philosophy  
  for solving problems}  
\journal{Prentice Hall}  
\pubyear{1984}  
\stoppublication
```

fermer

```
\startpublication[k=Bekke92,  
  t=article,  
  a=J.H.~ter~Bekke,  
  y=1992,  
  s=JB92]  
\artauthor[] {Johan} [J.H.] {ter} {Bekke}  
\arttitle{Semantic datamodeling}  
\journal{Prentice Hall}  
\pubyear{1992}  
\stoppublication
```

- Un grand nombre de champs sont disponibles et de nouveaux champs peuvent être créés. Cela est décrit dans la documentation du module bib.
- Les publications sont mises dans un fichier .bb1 de même nom que le fichier principal.

fermer

- Ce fichier `.bbl` peut soit être créé à la main, soit avec BibTeX grâce à un style `bst` fourni avec le module `bib`.
- Plusieurs bibliographies différentes sont accessibles depuis le même fichier.
- Une liste de publication est insérée avec `\placepublications`.
- Le tri est effectué par T_EXexec.



fermer

Gestion des polices : bas niveau

La gestion des polices dans ConTEXt est très différente de celle de L^ATEX.

- Pour modifier le style, il y a des commandes de bas niveau proches de celles de plain TEX : `\rm`, `\bf`, `\ss`, etc.
- Pour changer la taille, des commandes comme `\tfx`, `\tfixx`, `\tfa`, etc. permettent de choisir la taille *relativement* à la taille courante.

fermer

- `\em` permet l'emphase et inclut la correction italique.
- l'effet de toutes ces commandes peut être observé avec la commande `\showbodyfont` ; on peut par exemple écrire :

```
\showbodyfont [cmr]
```

ce qui donne

	[cmr]								\mr : Ag				
	<code>\tf</code>	<code>\sc</code>	<code>\sl</code>	<code>\it</code>	<code>\bf</code>	<code>\bs</code>	<code>\bi</code>	<code>\tfx</code>	<code>\tfxx</code>	<code>\tfa</code>	<code>\tfb</code>	<code>\tfc</code>	<code>\tfd</code>
<code>\rm</code>	Ag	AG	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\ss</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag
<code>\tt</code>	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag	Ag

fermer

Gestion des polices : haut niveau

Plus récemment, de nouveaux mécanismes ont été introduits, afin d'organiser les polices. Ces mécanismes sont les « *typescripts* » et ce ne sont vraiment que des groupes de définitions. Mais ces définitions peuvent être facilement sélectionnées via des *clés*.

Par exemple, en écrivant

```
\usetypescript[all] [computer-modern] [ec]
```

les définitions correspondantes du fichier `type-enc.tex` sont chargées. En voici quelques unes :

fermer

```
\starttypescript [all] [computer-modern] [ec]
\definefontsynonym [cmb10] [aeb10] [encoding=ec]
\definefontsynonym [cmbx10] [aebx10] [encoding=ec]
\definefontsynonym [cmbx12] [aebx12] [encoding=ec]
\definefontsynonym [cmbx5] [aebx5] [encoding=ec]
...
\stoptypescript
```

- Cela signifie que cmb10 est en fait un autre nom pour aeb10 et que cette dernière police est au codage T1 (ec).
- cmb10 n'est donc plus nécessairement un nom fixe de police, mais peut être redirigé vers d'autres polices.
- Les chaînes de synonymes (`\definefontsynonym`) sont sans limites.

fermer

- Des subdivisions plus fines peuvent être réalisés, par exemple en groupant les polices à empattements, bâtons, à chasse fixe et mathématiques :

```
\starttypescript [serif] [computer-modern] [default]
  \definefontsynonym [ComputerModern]          [cmr10]
  \definefontsynonym [ComputerModern-Italic]   [cmti10]
  \definefontsynonym [ComputerModern-Slanted]  [cmsl10]
  \definefontsynonym [ComputerModern-Bold]     [cmbx10]
  \definefontsynonym [ComputerModern-BoldItalic] [cmbxti10]
  \definefontsynonym [ComputerModern-BoldSlanted] [cmbxsl10]
  \definefontsynonym [ComputerModern-Caps]     [cmcsc10]
\stoptypescript
```

```
\starttypescript [sans] [computer-modern] [default]
  \definefontsynonym [ComputerModernSans]      [cmss10]
  \definefontsynonym [ComputerModernSans-Italic] [cmssi10]
  \definefontsynonym [ComputerModernSans-Slanted] [cmssi10]
  \definefontsynonym [ComputerModernSans-Bold]  [cmssbx10]
  \definefontsynonym [ComputerModernSans-BoldItalic] [cmssbx10]
```



fermer

```
\definefontsynonym [ComputerModernSans-BoldSlanted] [cmssbx10]
\definefontsynonym [ComputerModernSans-Caps] [cmss10]
\stoptypescript
...
```

- Les tailles disponibles sont définies dans un fichier `type-siz.tex` dont voici un échantillon :

```
\starttypescript [serif] [default] [size]
\definebodyfont
  [4pt,5pt,6pt,7pt,8pt,9pt,10pt,11pt,12pt,14.4pt,17.3pt]
  [rm] [default]
\stoptypescript
```

Chacune de ces définitions renvoie à d'autres définitions telles que :



fermer

```
\definebodyfont [12pt] [rm]
  [tf=cmr12,
   bf=cmbx12,
   it=cmti12,
   sl=cmsl12,
   bi=cmbxiti10 at 12pt,
   bs=cmbxsl10 at 12pt,
   sc=cmcsc10 at 12pt]
```

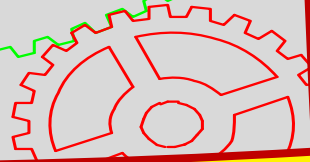
- La classification actuelle (nullement figée !) est donc proche de NFSS, mais il y a une notion de « collection » ou famille correspondant au groupe (serif, sans, mono, math). Une telle collection peut avoir un nom, par exemple `computer-modern`. Dans NFSS, ces quatre parties ne sont pas vraiment liées. Contrairement à NFSS, la graisse et l'aspect ne sont



fermer

pas par défaut des attributs orthogonaux. C'est le choix d'une collection, d'un style (serif, sans, mono, math) et d'une taille qui déterminent les significations de `\tf`, `\bf`, `\it`, etc. Comme on le voit, `\it` se situe au même niveau que `\bf`.

- La manipulation de ces définitions est un peu compliquée par le fait que les paramètres de `\usetypescript` n'ont pas une signification constante fonction de la position. Il s'agit

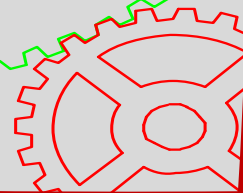


fermer

vraiment de paramètres fournissant un accès à une table.

- La taille globale d'un document peut être définie par

```
\setupbodyfont [12pt]
```



fermer

Polices : Affichage de code

Il est possible de définir des environnements type « verbatim » pour représenter des programmes, et obtenir automatiquement un *pretty-printing*. Par exemple, pour les programmes METAPOST, en écrivant :

```
\startMP
beginfig (12) ;
MyScale = 1.23 ;
draw unitsquare scaled MyScale
    shifted (10,20) ;
endfig ;
\stopMP
```



fermer

on obtient :

```
beginfig (12) ;  
MyScale = 1.23 ;  
draw unitsquare scaled MyScale  
  shifted (10,20) ;  
endfig ;
```



fermer

XML

Du code XML peut être inséré dans un fichier ConTeXt et ce code peut être traité moyennant des filtres. Voici un exemple de traitement de MathML :

```
\useXMLfilter[mml,mmp,mmc]

\startbuffer
<math>
  <apply> <eq/>
    <apply> <power/> <ci> a </ci> <ci> 2 </ci> </apply>
    <apply> <plus/>
      <apply> <power/> <ci> b </ci> <ci> 2 </ci> </apply>
      <apply> <power/> <ci> c </ci> <ci> 2 </ci> </apply>
    </apply>
  </apply>
</math>
```



fermer

```
\stopbuffer  
\processXMLbuffer
```

Le résultat est :

$$a^2 = b^2 + c^2$$

fermer

Autres aspects non évoqués

ConTEXt comporte de nombreuses autres facettes qui n'ont pas été évoquées :

- des modules de présentation permettent de réaliser des présentations comme celles-ci, avec des couches (*overlays*) ;
- la gestion de la micro-typographie avec l'algorithme HZ et les ponctuations pendantes (*character protruding*) ;
- un module de chimie, ppchTEX, basé sur PICTEX ;

fermer

- un module de représentation d'organigrammes (charts);
- un module pour écrire en chinois;
- etc.

fermer

Comment se mettre à ConTEXt

ConTEXt est un système très vaste et le risque est grand d'être découragé, surtout si l'on est un utilisateur averti et exigeant de L^ATEX. Voici quelques pistes :

- Lire l'article de Berend de Boer *L^ATEX in proper ConTEXt* qui fait un survol rapide des principales correspondances d'environnements entre L^ATEX et ConTEXt. Sa lecture est facile et conseillée.
- Feuilletter toutes les documentations mentionnées.

fermer

- Traduire un document L^AT_EX simple, pour lequel on n'a pas d'échéance. (Par la suite, on pourra éventuellement utiliser un script pour faire le gros de la traduction.)
- Poser des questions sur la liste `ntg-context@ntg.nl`, à laquelle on peut s'abonner depuis le site de PRAGMA. C'est une liste très active et les questions n'y sont jamais laissées sans réponse.

fermer

Conclusions : L^AT_EX ?

Faut-il abandonner L^AT_EX ?

- Non, L^AT_EX reste toujours utile pour un travail coopératif, ou pour certaines soumissions d'articles à des revues.
- Dans L^AT_EX il y a beaucoup de pièces de rechange, on sait que ça fonctionne et on sait à qui s'adresser en cas de problème.

fermer

Conclusions : pourquoi ConT_EXt ?

- pour ne pas rester enfermé dans L^AT_EX.
- pour avoir une perspective différente sur T_EX et donc avoir du recul.
- pour profiter de l'intégration de divers langages et donc se mettre à METAPOST, XML, ...
- pour tester des fonctionnalités expérimentales comme l'algorithme HZ.
- pour utiliser Γ (ConT_EXt avec Ω)

fermer

Aucun de ces points ne justifie à lui seul ConTExT,
mais c'est en ConTExT qu'ils sont tous réunis.

fermer

Conclusions : la suite

Que reste-t-il à faire en français ?

- le support français de ConTEXt doit encore être amélioré ;
- il faut rassembler les forces, écrire un manuel en français plus développé, etc.
- faire une page web pour les utilisateurs francophones avec tous les liens utiles.

fermer

Transparents

Merci !

fermer