

MAPS

NUMMER 25 • NAJAAR 2000



Voorzitter

Erik Frambach
erik.frambach@stelvio.nl

Secretaris

Maarten Gelderman
Vrije Universiteit Amsterdam
mgelderman@planet.nl

Penningmeester

Wybo Dekker
Servalys, Deil
wybo@servalys.hobby.nl

Bestuursleden

Frans Goddijn
frans@iaf.nl

Hans Hagen
PRAGMA ADE, Hasselt
pragma@wxs.nl

Postadres

Nederlandstalige T_EX Gebruikersgroep
Plantage Kerklaan 65/1
1018 CX Amsterdam

Postgiro

1306238
t.n.v. NTG, Deil
000-1662209-17
t.n.v. NTG, Veldegem, België

E-mail bestuur

ntg@ntg.nl

WWW

www.ntg.nl

Copyright © NTG

De **Nederlandstalige T_EX Gebruikersgroep (NTG)** is een vereniging die tot doel heeft het bevorderen van de kennis en het gebruik van T_EX. De NTG fungeert als een forum voor nieuwe ontwikkelingen met betrekking tot computergebaseerde document-opmaak in het algemeen en de ontwikkeling van 'T_EX and friends' in het bijzonder. De doelstellingen probeert de NTG te realiseren door onder meer het uitwisselen van informatie, het organiseren van conferenties en symposia m.b.t. T_EX en 'T_EX-producten'.

De NTG biedt haar leden onder andere het volgende:

- Tweemaal per jaar een NTG-bijeenkomst.
- Tweemaal per jaar het uitgebreide NTG-tijdschrift MAPS.
- De 'T_EX Live CD-ROM' voor Unix-gebruikers en de 'CTAN CD-ROM' met de complete T_EX software-archieven.
- Speciale MAPS-uitgaven (T_EX-cursus, FAQ, en PR-set).
- Verschillende discussielijsten (mailing lists) over T_EX-gerelateerde onderwerpen, zowel voor beginners als gevorderden, algemeen en specialistisch.
- De FTP server ftp.ntg.nl waarop vele honderden megabytes aan algemeen te gebruiken 'T_EX-producten' staan.
- De www server www.ntg.nl waarop algemene informatie staat over de NTG, bijeenkomsten, publicaties, en links naar andere T_EX sites.
- Korting op (buitenlandse) T_EX-conferenties en -cursussen en op het lidmaatschap van andere T_EX-gebruikersgroepen.

Lid worden kan door overmaking van de verschuldigde contributie naar de NTG-giro (zie links); daarnaast dient via www.ntg.nl een informatieformulier te worden ingevuld. Zodat nodig kan ook een papieren formulier bij het secretariaat worden opgevraagd.

De contributie bedraagt *f*90 (BF 1660); voor studenten geldt een tarief van *f*60 (BF 1100). Dit geeft *geen stemrecht* en een bewijs van inschrijving is vereist. Een gecombineerd NTG/TUG-lidmaatschap levert een korting van 10% op beide contributies op. De prijs in gulden wordt bepaald door de dollarkoers aan het begin van het jaar. De ongekorte TUG-contributie is momenteel \$65.

T_EX is een door professor Donald E. Knuth ontwikkelde 'opmaaktaal' voor het letterzetten van documenten, een documentopmaakstelsel. Met T_EX is het mogelijk om kwalitatief hoogstaand drukwerk te vervaardigen. Het is eveneens zeer geschikt voor formules in wiskundige teksten. Er is een aantal op T_EX gebaseerde producten, waarmee ook de logische structuur van een document beschreven kan worden, met behoud van de letterzetmogelijkheden van T_EX. Voorbeelden zijn L^AT_EX van Leslie Lamport, $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX van Michael Spivak, en ConT_EXt van Hans Hagen.

Inhoudsopgave

Nummer 25, najaar 2000

- Redactioneel, *Sven Bovin en Siep Kroonenberg* **1**
- Euro \TeX , *Erik Frambach* **2**
- De NTG en het Internet, *Jules van Weerden* **4**
- \TeX user groups, *Erik Frambach en Siep Kroonenberg* **6**
- Nieuws van CTAN, *Piet van Oostrum* **9**
- Con \TeX t en pdftex, *Berend de Boer* **13**
- Gezeefd uit de TEX-NL discussielijst, *Frans Goddijn* **19**
- Toolbox: een syllabus, *Maarten Gelderman* **56**
- The \TeX Live CDROM, easy and fast installation, *Wybo Dekker* **72**
- pdf \TeX 's little secret; tracking positions, *Hans Hagen* **74**
- Typesetting pdf annotations, *Hans Hagen* **79**
- Hanging punctuation, a pdf \TeX microtypographic extension, *Hans Hagen* **81**
- Antykwa Półtawskiego:, a parameterized outline font, *Bogusław Jackowski, Janusz M. Nowacki and Piotr Strzelczyk* **86**
- A macro routine for writing text along a path in MetaPost, *Santiago Muelas* **103**
- MetaFun chapter 3: embedded graphics, *Hans Hagen* **114**

Dit is alweer het vijftiengste nummer van MAPS (en wederom later dan onze bedoeling was). Een nummer met een gevarieerde inhoud. Veel aandacht voor grafische en typografische zaken: fonts, typografie, METAPOST, het komt allemaal aan bod.

Traditiegetrouw heeft Hans Hagen weer een aantal interessante bijdragen geleverd rond pdf \TeX en METAPOST. Onze Spaanse collega Santiago Muelas presenteert een METAPOST-routine voor het zetten van tekst op een willekeurig pad.

Uit Polen komt een artikel over het ontwerpen van een geparametriseerd outline font, een overname uit de Proceedings van Euro \TeX 99.

De alledaagse praktijk komt ruimschoots aan bod in een dubbele aflevering van *Gezeefd* van de hand van Frans Goddijn – terug van weggeweest, en met een nieuwe, asymmetrische layout. Maarten Gelderman doet in de Toolbox uit de doeken hoe hij een groot \TeX -project heeft aangepakt en wat voor gereedschap hij daarvoor gebruikt. Truiks en scripts worden dit maal geleverd door Wybo Dekker met een stukje over de installatie van de \TeX Live CDROM op de harde schijf.

Wat nieuws betreft, in de eerste plaats de aankondiging van Euro \TeX 2001, ditmaal door het NTG te organiseren. Ook twee nieuwe rubrieken, waarvan het de bedoeling is dat zij een vast onderdeel zullen worden van de MAPS: Berend de Boer brengt een nieuwsrubriek over Con \TeX t en pdf \TeX , waarin de laatste ontwikkelingen rond dit macropakket en dit back-end zullen worden besproken, en Piet van Oostrum bespreekt nieuwe aanwinsten in het CTAN-archief.

Verwacht: een nieuwe PR-set

Tijdens de ledenvergadering in juni is de NTGPR-set ter sprake gekomen. De huidige PR-set is aan vernieuwing toe. Zo'n PR-set moet aan de ene kant een showcase zijn voor wat er allemaal kan met \TeX , en aan de andere kant mooie voorbeelden bevatten van hoe \TeX in de praktijk wordt gebruikt. We denken ook aan de MAPS-CD en een speciaal nummer van de MAPS als onderdeel van de PR-set.

Er is inmiddels wat materiaal binnengekomen, en er is heel wat bestaand materiaal dat goed in de PR-set zou passen, maar we moeten er ook een keertje echt onze schouders onder zetten, en we hebben het natuurlijk allemaal druk met andere dingen.

Kleur

Ook al in het kader van de PR-set vinden we het tijd worden voor een nummer in kleur. We twijfelen er niet aan dat Hans Hagen ons op iets moois zal kunnen vergasten maar laat dat andere auteurs niet ervan weerhouden om ook hun beste beentje voor te zetten!

Bemanning

Sven Bovin is nu de nieuwe hoofdredakteur, en productie was in handen van Siep Kroonenberg. Taco Hoekwater is teruggetreden als hoofdredakteur, en het voelt beslist onwennig om zonder zijn stuwende kracht een MAPS in elkaar te zetten.

Colofon

De MAPS wordt gezet met gebruikmaking van een \LaTeX class file en een Con \TeX t module. De gebruikte fonts zijn Adobe Times-Roman+Old Style Figures en Frutiger, met een versmalde Courier voor de 'verbatim' omgevingen. Compilatie gebeurt met web2c versie 7.3.2 onder Debian Linux 2.2, en we drukken op 90-grams coated papier vanaf een PDF bestand dat wordt gemaakt met pdf[e]tex 0.14 en dvips versie 5.86, gedistilleerd met Adobe's Distiller 4.05 draaiend onder Windows 95.

eurotex

EuroT_EX2001 conference

Erik Frambach
erik.frambach@stelvio.nl

abstract

Announcement of the 12th European T_EX conference, which will be held in Kerkrade in the Netherlands, from 23 to 27 September 2001. The theme is “T_EX and Meta: the Good, the Bad and the Ugly Bits”.

keywords

EuroT_EX, conference

T_EX and Meta:

the Good, the Bad and the Ugly Bits

That’s the theme of the 12th European T_EX conference, which will be held in Kerkrade in the Netherlands, from 23 to 27 September 2001.

So, what are “Good, Bad and Ugly Bits”? Traditionally we tend to focus on Good Bits because they are so rewarding. It feels good to see a beautifully typeset document that you were able to produce using all the power that T_EX and Metafont/MetaPost provide. We try to forget how many Bad Bits we had to overcome, and are proud that we found more or less elegant ways to program around them. Nevertheless, some of these difficulties should be regarded as Ugly Bits that would not exist in an ideal world. Well then what can we do?

- First of all, keep up the Good Bits and extend them if possible.
- Analyze the Bad Bits, learn from them, and find easy and generic ways to get around them.
- Find the Ugly Bits and eradicate them!

Let’s make progress! Of course we need a lot of discussion to distinguish between Good, Bad and Ugly, and about ways to deal with them. We invite you all to demonstrate or discuss either of these bits. Feel free to present your application (Good and/or Bad and/or Ugly) at the conference!

Call for papers

If you would like to present a paper at the conference, please submit your proposal to the Program Committee, eurotex-pc@ntg.nl. We welcome submissions in T_EX, L^AT_EX, plain T_EX, ConT_EXt or other T_EX macro packages.

Program

As this moment, one year before the start of the conference, we cannot give you any realistic idea of the program yet. We will publish the program as soon as possible at the EuroT_EX2001 website: <http://www.ntg.nl/eurotex/>.

Registration

In contrast to most other T_EX conferentias for EuroT_EX2001 we charge a flat fee that includes just about everything. No need to find a (much more expensive) hotel, no need to find a place to have lunch or dinner. It’s all arranged for you at the beautiful Rolduc site, so you get every opportunity to focus on T_EX and meet all the people you’ve been wanting to talk to. The fee includes:

- Lodging on Sunday, Monday, Tuesday and Wednesday on conference site
- Breakfast and lunch and dinner on Monday and Tuesday and Wednesday
- Conference proceedings
- A special social event in Maastricht

Note that the fee rises as the conference date approaches. The earlier you register the less you pay. However, you are always allowed to pay more :-). In that case your *name* will be listed as *sponsor* of the conference.

If you register early (i.e. before 1 January 2001) you will pay only 350 Euro. After 1 January the fee rises to 450 Euro; after 1 June it rises to 550, and after 1 August it rises to 650 Euro. Note that if you are *not* a member of any listed T_EX users group you will be charged an extra 150 Euro, so it’s really much more profitable to join NTG, TUG, Dante, UKTUG, or any of the other user groups. Actually, it’s a good idea to join a T_EX users group anyway! See the web page <http://www.tug.org/lugs.html> for a listing of currently active T_EX users groups.

Sponsoring

In case you or your company/institute would like to sponsor the conference in any way, please contact the Organizing Committee (eurotex-org@ntg.nl) and visit our sponsor pages at <http://www.ntg.nl/eurotex/>. There are many options for sponsoring, and we will do our best to make it worth your while.

Accommodation

Attending an event at Rolduc Conference Centre means entering historic surroundings. Located near the Limburg town Kerkrade, the imposing abbey of Rolduc was founded in 1104 by Ailbertus van Antoining. For centuries it was an Augustinian monastery, later becoming a seminary.

Since the twelfth century, Rolduc has been a centre for culture, scholarship and religion. Evidence can be found in the exquisite abbey church itself, the crypt, and the 18th-century Rococo library, which helps create Rolduc's unique atmosphere. It is to this historic site that visitors now come from all over the world.

While retaining its historic ambience, Rolduc has been transformed into an up-to-the-minute centre for conferences and other events. After a full-scale restoration and renovation lasting many years, this historic monument has been restored into its former glory, as visitors can see for themselves during one of the guided tours.

All the facilities are state-of-the-art, with the latest audiovisual equipment and other aids. Surrounded by a splendid park, and with plenty of parking space, Rolduc Conference Centre's unique atmosphere and the hospitality of its staff guarantee the tranquillity, concentration and relaxation you need to ensure that your event is a success.

Rolduc's splendid wooded grounds, with their gardens and ponds, the athletics field, tennis court and other facilities, ensure that guests have everything they need when it comes to sport and recreation. In addition, guests can enjoy a drink in one of the cellar bars with its vaulted ceilings, or on a terras in one of the lovely walled gardens.

Rolduc has hotel rooms, located throughout the complex, which can accommodate one or more guests. Hotel rooms are sober but very efficient.

The Limburgian people's nature is to spoil their guests, and Rolduc Congress Centre takes pride in continuing that tradition. The cuisine is uncomplicated and very tasty.

How to get there

Rolduc is located just outside of Kerkrade, which is a city about 30 km east of Maastricht and about 20 km north of Aachen (Germany). For more details see the EuroT_EX2001 website (<http://www.ntg.nl/eurotex/rolduc.html>).

The Challenge

The conference is scheduled for about a year from now, so meanwhile here's a challenge for you: make a visual expression of "the Good, the Bad and the Ugly Bits", using T_EX and Meta. The winner of the challenge will be awarded! The rules are:

- Your artwork must be T_EX and/or Metafont/MetaPost based;
- You may use other tools as well for minor functions;
- Your artwork must be original;
- Your artwork must be printable on a single A4 page;
- Your artwork must be made available in a format suitable for publication on the EuroT_EX 20001 web pages;
- You may submit as many artworks as you like;
- The artwork will be exhibited during the conference;
- Non-participants of the conference can submit, but cannot win;
- All participants of the conference can vote for the best work of art;
- The winner will be awarded!

Please submit your artwork to eurotex-art@ntg.nl.

Organization

If you have any questions or remarks concerning the conference, please contact the EuroT_EX2001 Organizing Committee at eurotex-org@ntg.nl.

Please contact the Program Committee (eurotex-pc@ntg.nl) if you have any questions about giving a presentation at the conference.

We hope to see you all in Rolduc next year!

internet

De NTG en het Internet

Jules van Weerden
email: Jules.vanWeerden@let.uu.nl
mmv Maarten Gelderman
email: mgelderman@bik.econ.vu.nl

abstract

De NTG is niet alleen met webpagina's op het Internet aanwezig, maar ook middels een aantal discussielijsten. In dit artikel wordt kort aangegeven wat een discussielijst is, hoe je je aan- en afmeldt en wat je te wachten staat na aanmelding. Tevens wordt een overzicht gegeven van de in Nederland aanwezige T_EX-gerelateerde lijsten.

keywords

discussielijsten, tex-nl

Het bekendste onderdeel van het Internet is ongetwijfeld het World Wide Web. In deze bijdrage wil ik het hebben over discussielijsten op het Internet. Eerst zal ik globaal een beeld schetsen van wat zo'n lijst is en hoe een discussielijst werkt. Vervolgens kom ik toe aan het bespreken van de Nederlandse T_EX-gerelateerde discussielijsten.

Discussielijsten

Veel mensen zijn zich niet bewust van het feit dat er discussielijsten bestaan. De reden hiervoor is eenvoudig. In tegenstelling tot bij voorbeeld het World Wide Web of ftp heb je voor het gebruik van discussielijsten geen aparte software nodig. Een email-programma (en natuurlijk toegang tot Internet) volstaan. De werking van een discussielijst zelf is eigenlijk te simpel voor woorden. Een discussielijst is voor de gebruiker niets anders dan een email-adres, bij voorbeeld `discussielijst@lijstencomputer.nl`. Alle mail die naar dit email-adres wordt gestuurd wordt automatisch doorgestuurd naar alle leden van de discussielijst. Het email-adres van de belangrijkste lijst in Nederland, `tex-nl`, is bij voorbeeld `tex-nl@nic.surfnet.nl`. Alle mail die naar dit adres wordt gestuurd komt bij alle 183 leden aan.¹ Op de lijstcomputer draait een softwarepakket (LISTSERV en MAJORDOMO zijn de populairste pakketten) dat er voor zorgt dat dit automatisch gebeurt. De op deze wijze gecreëerde lijst kan gebruikt worden om over van alles en nog wat te discussiëren. De T_EX-lijsten hebben veelal een vraag- en antwoord-

karakter. De leden van deze lijsten stellen op de lijst vragen over T_EX-gerelateerde problemen en de andere leden (en met name Piet van Oostrum die hiermee terecht een erelidmaatschap van de NTG heeft verdiend) proberen deze vragen te beantwoorden.

Tot zover erg aardig natuurlijk, maar dan moet een lijst wel leden hebben. Ook dit proces is geautomatiseerd. Naast het adres `discussielijst@lijstencomputer.nl` bestaat er een adres waarmee het programma direct benaderd kan worden, bij voorbeeld `listserv@lijstencomputer.nl`. Het is niet de bedoeling dat er naar dit adres gewone berichten worden gestuurd, daar kan de software niets mee. In plaats daarvan moet de mail korte commando's bevatten. Je kan bij voorbeeld een mailtje met als inhoud

```
HELP  
END
```

versturen naar `listserv@nic.surfnet.nl`, je krijgt dan een mailtje met uitleg over de beschikbare commando's terug.² Om je aan te melden als lid van bij voorbeeld `tex-nl`, kan je het volgende mailtje versturen naar `listserv@nic.surfnet.nl`:

```
SUBSCRIBE TEX-NL Voornaam Achternaam  
END
```

De LISERV-software kan vervolgens drie dingen doen.

1. Een mailtje terugsturen met de mededeling dat je geen lid kunt worden. Bij `tex-nl` is het uiterst onwaarschijnlijk dat dit gebeurt, maar bij andere lijsten (b.v. de bestuurslijst van de NTG) kan dit voorkomen.
2. Een mailtje terugsturen met de mededeling dat je lid bent geworden van de lijst. In dit mailtje wordt tevens uitgelegd hoe je je weer af kunt melden. Bewaar het!

1. Dit is niet helemaal waar. Indien het mailtje door één van de 183 leden wordt verstuurd, komt het normaal gesproken alleen bij de overige leden aan. Wil je het ook zelf weer terughebben stuur een melding aan het beheersadres (zie verder) met als commando `'repro <lijstnaam>'`. Dit is een faculteit van LISERV. Bij de MajorDomo lijsten krijgt iedereen het bericht; ook de afzender. Op sommige lijsten is het bovendien alleen aan leden van de lijst of soms zelfs alleen aan een lijstbeheerder toegestaan om mail te verzenden.

2. Helaas zijn de commando's voor LISERV en MAJORDOMO niet identiek.

3. Een mailtje terugsturen met de mededeling dat je voor de zekerheid nog even moet bevestigen dat je echt lid wilt worden van de lijst. Dit is met name om te voorkomen dat ongeldige emailadressen aan een discussielijst worden toegevoegd en om te zorgen dat mensen niet ongewenst aan een lijst worden toegevoegd.

Na deze eenvoudige actie ben je lid van `tex-nl` en kan je mail (met vragen over \TeX en \LaTeX) naar deze lijst versturen. Tevens ontvang je vanzelfsprekend alle mail die anderen naar deze lijst zenden (gemiddeld zo'n 25 mailtjes per week). Om je weer af te melden stuur je een mailtje met het commando `UNSUBSCRIBE TEX-NL` naar de listserv.

Lijsten in Nederland

Ook op het gebied van de elektronische berichtenuitwisseling gaat het goed met de NTG. Op dit moment is een zevental lijsten geïnitieerd vanuit Nederland (voor zover ik weet ;-).

- tex-nl** Algemene discussielijst over \TeX in Nederland.
- 4tex** Lijst voor vragen en mededelingen betreffende het 4TeX -systeem van Erik Frambach en Wietse Dol.
- ntg-tex-tools** Vragen-, antwoorden- en opmerkingenlijst over allerlei programmatuur die gebruikt kan worden om de resultaten van \TeX nog beter te maken. Of zelfs om \TeX te genereren
- ntg-context** Discussie lijst over ConTeXt, een parameter-gestuurd \TeX macro pakket
- ntg-ppchtex** Deze lijst gaat over PPCH \TeX , een \TeX macro pakket dat gebruikt kan worden om chemische structuurformules te zetten
- ntg-toekomsttex** Discussielijst over de toekomst van \TeX

De eerste twee lijsten zijn te gast op de email-server `LISTSERV` van `SURFNET`. De lijsten die beginnen met 'ntg-' zijn te gast bij de MajorDomo email-server van de faculteit der Letteren in Utrecht.

Verder zouden nog de lijsten genoemd kunnen worden:

- teTeX** lijst, waarop gediscussieerd wordt over het totaalpakket dat Thomas Esser heeft samengesteld van de basis-programmatuur die je nodig hebt om \TeX op o.a. UNIX te draaien. Is de basis van de \TeX -live CD.
- CTAN-Ann** De beheerder(s) van `CTAN` gebruiken deze lijst voor de aankondigingen van nieuwe bestanden

op `CTAN`. Handig om op de hoogte te blijven van het uitkomen van de nieuwste macro's en programma's. Je kunt er zelf (dus) niet posten.

Op alle lijsten wordt levendig gediscussieerd. Voor de lijsten `ctan-ann`, `tex-nl`, `4TEX` en `teTeX` houd ik (JvW) zelf een archief bij en omdat het niet erg geheim is, kan iedereen daar ook bij via de ftp-server `ftp.let.uu.nl` in de directory: `/pub/tex/<lijst>` [lijstnamen in kleine letters].

Let verder op het verschil tussen de `LISTSERV`- en de MajorDomo-software. Voor de eerste kun je op de regel 'subscribe <lijstnaam>' ook nog een willekeurige tekst toevoegen. Bv

```
subscribe tex-nl Jules van Weerden, CIM, Fac \
der Letteren, UU, NL
```

De vrije tekst wordt als commentaar in de verzendlijst opgenomen en door de programmatuur genegeerd. Bij MajorDomo mag die extra informatie er NIET staan. Die wordt nl beschouwd als het email adres waar de berichten heen moeten.

Als je je voor een lijst wil aanmelden moet je een email-bericht sturen aan het bijbehorende adres. Het onderwerp is niet echt van belang, maar je krijgt het in de reactie terug, dus je kunt het er aan herkennen. In de tekst van het bericht neem je de regel op: `subscribe <lijst>`.

De verschillende aanmeld-adressen:

- Voor `tex-nl` en `4TEX`: `LISTSERV@nic.surfnet.nl`.
- Voor `ntg-tex-tools`, `ntg-ppchtex`, `ntg-context` en `ntg-toekomsttex`: `Majordomo@ntg.nl`.
- voor `teTeX`: `Majordomo@informatik.uni-hannover.de`
- voor `CTAN-Ann`: `listserv@urz.uni-heidelberg.de`

De reden dat de lijsten met 'ntg-' beginnen is dat we (nog) geen scheiding hebben aangebracht tussen de lijsten van de faculteit der Letteren en de lijsten van de NTG. Hopelijk kunnen we binnenkort alles splitsen. Dan krijg je ook de reactie terug van `Majordomo@ntg.nl` en niet zoals nu van `Majordomo@let.uu.nl`.

Voor alle lijsten geldt dat als je een probleem met de lijst hebt dat je een bericht kunt sturen aan `owner-<lijst>@<email-server>` waar een 'echt' persoon achter zit, die kan helpen met het oplossen van het probleem.

Geniet van de lijsten en bestrijd de SPAM (ongewenste reclame).

user groups

TeX user groups around the world

Erik Frambach
email: erik.frambach@stelvio.nl
Siep Kroonenberg
email: siepo@cybercomm.nl

All over the world TeX users have formed networks of *user groups* on an informal basis. These user groups consist of enthusiastic TeX users who share their problems and solutions with anyone who wants to join the TeX community. They usually communicate through email and often produce printed journals with contributions from members.

The journals and discussions on email are essential for users who want to be informed about the latest developments. Email is important for getting information on, e.g. what is the newest version of program *x*, where do I find a macro for problem *y*, how do I install program *z*. For this purpose, a list of frequently asked questions ('FAQ') is maintained, and is distributed regularly.

Below we have listed all TeX user groups currently known to us. Much of this information comes from the list of user groups on the TUG website (<http://www.tug.org/lugs.html>).

New since the last MAPS issue in this listing are the **Chinese TeX Users Group** and the **Vietnamese TeX Users Group**.

AsTeX (French-speaking)

Michel Lavaud, President
Association pour la diffusion de logiciels scientifiques liés à TeX
Association AsTeX
BP 6532
45066 Orleans cedex 2
France
tel: +33 2 38 64 09 94
email: astex-admin@univ-orleans.fr
discussion list astex@univ-orleans.fr

CervanTeX (Spanish-speaking)

Grupo de Usuarios de TeX Hispanohablantes
Association formed in February 1999
José Ra Portillo Fernández:
Departamento de Matemática Aplicada I
Escuela Técnica Superior de Arquitecturas
Avenida de la Reina Mercedes, 2
E-41012 Sevilla

Spain

Public mailing list: spanish-tex@eunet.es
(Send subscription requests to this list)
email: josera@gordo.us.es

Chinese TeX Users Group

Suite S206, South Building
No. 80, Xue Yuan, South Road
HaiDian District
Beijing 100081
China
tel: +86 10 62186674
email: info@mail.rons.net.cn
WWW page: http://www.rons.net.cn/english/tech_corner/texclub.html

CSTUG (Czech and Slovak Republics)

Petr Sojka, President
Československé sdružení uživatelů TeXu
CsTUG, c/o FI MU
Botanická 68a
CZ-602 00 Brno
Czech Republic
tel: +420 5 41212352
fax: +420 5 41212568
email: cstug@cstug.cz
WWW page: <http://www.cstug.cz/>
journal: *Zpravodaj CSTUG*

CyrTUG (Russia)

Eugenii V. Pankratiev, President
Irina Makhovaya, Executive Director
Associaciia Pol'zovatelej Kirillicheskogo TeX'a
Mir Publishers
2, Pervyj Rizhskij Pereulok
Moscow 129820
Russia
tel: +7 95 286 0622, 286-1777
fax: +7 95 288 9522
email: cyrtug@cemi.rssi.ru
WWW page: <http://www.cemi.rssi.ru/cyrtug/>

DANTE e.V. (German-speaking)

Thomas Koch, President
Deutschsprachige Anwendervereinigung TeX e.V.
Postfach 101840

D-69008 Heidelberg
 Germany
 tel: +49 6221 29766
 fax: +49 6221 167906
 email: dante@dante.de
 WWW page: <http://www.dante.de/>
 journal: *Die T_EXnische Komödie*

DK-TUG (Danish)
 Thomas Widman, President
 DK-TUG
 Department of Mathematical Sciences
 University of Aarhus
 Ny Munkegade Building 530 DK-8000 Aarhus
 Denmark
 email: dk-tug-bestyrelse@sunsite.auc.dk

Estonian User Group
 Enn Saar, Tartu
 Astrophysical Observatory, Toravere
 EE 2444 Estonia
 email: saar@aai.ee

εφτ (Greek speaking)
 Greek T_EX Friends Group
 Apostolos Syropoulos, President
 366, 28th October Str.
 GR-671 00 Xanthi
 Greece
 tel: +30 541 28704
 email: apostolo@obelix.ee.duth.gr
 WWW page: <http://obelix.ee.duth.gr/εφτ/>

Grupo de Utilizadores de TeX (Portuguese)
 No formal user group yet
 For information, contact Pedro Quaresma de Almeida
 Email: pedro@mat.uc.pt

GUST (Poland)
 Tomasz Przechlewski (President)
 Polska Grupa Użytkowników Systemu T_EX
 Instytut Matematyki Uniwersytetu Gdańskiego
 ul. Wita Stwosza 57
 80 - 952 Gdańsk
 Poland
 email: ekotp@univ.gda.pl
 WWW page: <http://www.gust.org.pl/>
 journal: *GUST*

GUTenberg (French-speaking)
 Michel Goossens, President
 Groupe francophone des Utilisateurs de T_EX
 Association GUTenberg

c/o Irisa, Campus universitaire de Beaulieu
 F-35042 Rennes cedex
 France
 tel/fax: +33 1 30 87 06 25
 email: gut@irisa.fr
 WWW page: <http://www.gutenberg.eu.org/>
 journal: *Les Cahiers GUTenberg*

ITALIC (Irish)
 No formal user group yet
 Public mailing list: ITALIC-L@listserv.heanet.ie
 (send subscription requests to
 listserv@listserv.heanet.ie)
 Peter Flynn
 Computer Centre
 University College
 Cork
 Ireland
 email: pflynn@www.ucc.ie

Lietovos TeX'o Vartotojų Grupė (Lithuanian T_EX
 Users Group)
 Vytas Statulevičius, Chair
 Akademijos 4
 LT-2600 Vilnius
 Lithuania
 tel: +370 2 359 609
 fax: +370 2 359 804
 email: vytass@ktl.mii.lt
 WWW page: <http://www.vtex.lt/tex/>

Nordic TeX User Group (Scandinavian countries)
 Dag Langmyhr, Chair
 Nordic T_EX Users Group
 Department of Informatics
 PO Box 1080 Blindern
 University of Oslo
 N-0316 Oslo
 Norway
 tel: +47 22 85 24 50
 fax: +47 22 85 24 01
 email: dag@ifi.uio.no
 WWW page: <http://www.ifi.uio.no/~dag/ntug/>

NTG (Dutch-speaking)
 Erik Frambach, Chair
 Nederlandstalige T_EX Gebruikersgroep
 Plantage Kerklaan 65/1
 1018 CX Amsterdam
 The Netherlands
 email: ntg@ntg.nl
 WWW page: <http://www.ntg.nl/>
 journal: *MAPS*

T_EXCeH (Slovenian T_EX User Group)

Vladimir Batagelj
 Jadranska 19
 SI-61111 Ljubljana
 Slovenia
 email: Tex.Ceh@fmf.uni-lj.si
 WWW page: <http://vlado.fmf.uni-lj.si/texceh/texceh.htm>

Tirant lo T_EX (Grup d'usuaris catalanoparlants de T_EX)

Gabriel Valiente Feruglio
 Technical University of Catalonia
 Department of Software
 Mòdul C6, Campus Nord
 Jordi Girona Salgado, 1-3
 E-08034 Barcelona
 Catalonia
 email: valiente@lsi.upc.es
 Mailing list: catala-tex@aligna.cesca.es (send
 subscription requests to listserv@cesca.es)
 WWW page: <http://www-lsi.upc.es/~valiente/tug-catalan.html>

TUG (International user group, LUG for T_EX users not covered by some other group)

Mimi Jett, President
 T_EX Users Group
 1466 NW Front Avenue, Suite 3141
 Portland, OR 97209
 USA
 tel: +1 503 223 9994
 fax: +1 503 223 3960
 email: office@tug.org
 WWW page: <http://www.tug.org/>
 journal: *TUGboat*

TUGIndia (Indian)

Prof. (Dr.) K. S. S. Nambooripad, Chairman
 Indian T_EX Users Group
 Kripa, TC 24/548, Sastha Gardens

Thycaud, Trivandrum 695014

India
 tel: +91 471 324341
 fax: +91 471 333186
 email: tugindia@river-valley.com
 WWW page: <http://www.river-valley.com/tug/>

TUG-Philippines (Philippines T_EX Users Group)

Dr. Felix P. Muga II, President
 Mathematics Department
 Ateneo de Manila University
 Loyola Heights, Quezon City
 Philippines
 Tel: +63 2 426 6001 local 2515
 Fax: +63 2 426 6008
 email: fpmuga@admu.edu.ph and
fpmuga@philonline.com

UK TUG (United Kingdom)

Philip Taylor, Chairman
 UK T_EX Users' Group
 For information:
 Peter Abbott
 1 Eymore Close
 Selly Oak
 Birmingham B29 4LB
 England
 email: uktug-enquiries@tex.ac.uk
 WWW page: <http://uk.tug.org/>
 journal: *Baskerville*

ViêtTUG (Vietnamese T_EX Users Group)

Nguyen-Dai Quý LTAS-University of Liège
 Rue des Chevreuils, 1, Bât B52, Local 522B
 B4000, Liège
 Belgium
 tel: +32 4 366 9098
 Fax: +32 4 366 9311
 email: viettug@iris.ltas.ulg.ac.be
 WWW page: <http://iris.ltas.ulg.ac.be/viettug/>

Nieuws van CTAN

Een uittreksel uit de recente bijdragen in het CTAN archief

Piet van Oostrum

abstract

Dit artikel beschrijft een aantal recente bijdragen uit het CTAN archief. Het is de bedoeling om in elke MAPS zo'n artikel te plaatsen. Ik zal hierbij één of meer bijdragen wat uitvoeriger beschrijven en andere bijdragen kort noemen. De selectie is gebaseerd op wat ik zelf interessant vind en wat ik denk dat voor veel anderen interessant is. Het is dus wel een persoonlijke keuze. Het heeft niet de bedoeling om een volledig overzicht te geven. De uitgebreidere bijdragen zijn ook geen handleidingen. Beschouw het maar als een soort menukaart die de bedoeling heeft om de lezer te laten watertanden.

keywords

T_EX, L^AT_EX, packages, CTAN, proceedings

Samenstellen

Een van de problemen die organisatoren van conferenties (zoals de komende EuroT_EXconferentie) tegenkomen is het samenstellen van de *proceedings* van de conferentie. De individuele artikelen waaruit de proceedings bestaan worden meestal door de auteurs aangeleverd en de redacteurs mogen er een boekwerk van maken. Een soortgelijk probleem komt voor bij tijdschriften, zoals onze MAPS. Het grote probleem is dat de bijdragen in het geheel moeten passen. Elke auteur kent slechts zijn eigen deel en wil dit zoveel mogelijk als één geheel behandelen. De redacteurs zien echter het gehele document. We gaan er in het volgende vanuit dat de individuele artikelen in L^AT_EX gemaakt zijn.

Er zijn een aantal manieren om dit probleem aan te pakken. De meeste daarvan heb ik in de praktijk toegepast zien worden.

□ Het meest rigoureuze is om van de auteurs te eisen dat ze zich aan voorgeschreven beperkingen in hun T_EX/L^AT_EX code houden. Ze moeten dan een speciale *documentclass* gebruiken en geen eigen commando's verzinnen. De class is zo ontworpen dat het gemakkelijk ingepast kan worden in het totale document.

□ Druk alle artikelen individueel af zonder paginanummers of headers/footers¹. Voeg aan het totale document nieuwe paginanummers toe.

Het toevoegen van de paginanummers kan enige hoofdbrekens kosten. Om problemen met T_EX en/of L^AT_EX te voorkomen is de eenvoudigste manier om dit maar buiten T_EX om te doen. Bijvoorbeeld door de stapel afgedrukte artikelen nog eens in de laserprinter te stoppen en een document met alleen paginanummers af te drukken. Sommige printers kunnen er niet zo goed tegen als het papier er nog eens doorheen gehaald wordt. Dit komt doordat de structuur van het papier door het afdrucken verandert. Ook kan het voorkomen dat er een storing optreedt waardoor de job verknald wordt. Opnieuw beginnen is dan het enige alternatief.

Wanneer het maken van een 'camera-ready' document geheel in eigen beheer wordt gedaan kan deze oplossing heel handig zijn. Een extra voordeel is dat we ook documenten kunnen gebruiken die met andere software gemaakt is. Afgezien van de boven beschreven problemen die echter tijdelijk van aard zijn, zijn er weinig problemen te verwachten. Wanneer we echter elektronische documenten moeten aanleveren bij een drukkerij is dit geen oplossing.

□ Een variant die wel een correct elektronisch document oplevert is die waarbij de artikelen zonder paginanummers tot bijvoorbeeld een Postscript of PDF file samengevoegd worden. Dit kan bij Postscript o.a. door met het programma *dviconcat* de afzonderlijke DVI files samen te voegen en dan met *dvips* een postscript file te maken. Of bij gebruik van PDF kunnen we de afzonderlijke artikelen omzetten naar PDF en ze dan met Acrobat aan elkaar plakken. Rest het probleem om er paginanummers aan toe te voegen. Voor Acrobat zijn er uitbreidingen te verkrijgen die dat doen. Die kunnen zelfs bestaande paginanummers weghalen. Met Postscript wordt het wat moeilijker maar als alle Postscript uit *dvips* komt is het niet zo moeilijk om een Perl of Python script te maken die de paginanummers erbij zet².

1. Wanneer ik in het vervolg "paginanummers" schrijf bedoel ik daarbij ook eventuele headers en/of footers

2. Netscape op Unix systemen drukt geen paginanummers af wanneer

□ Wanneer alle artikelen als \TeX of \LaTeX aangeleverd worden is het ook niet zo moeilijk om per artikel de paginanummers aan te passen en ze individueel te verwerken. Vaak zijn er ook nog extra aanpassingen nodig om de titels uniform te krijgen, maar dat geldt ook voor de andere oplossingen.

□ De hier te beschrijven documentclass **combine** heeft als doel om de individuele artikelen binnen een omvattend document te gebruiken waarbij de artikelen in de meeste gevallen niet aangepast hoeven te worden. In normaal \LaTeX gebruik is dit niet mogelijk omdat er in de document slechts één `\documentclass` commando mag voorkomen en slechts één document omgeving. De **combine** class zal dus moeite moeten doen om de binnenste voorkomens hiervan (in de afzonderlijke artikelen) en ook de daarin voorkomende `\usepackage` commando's onschadelijk te maken.

Combine

De **combine** documentclass van Peter Wilson maakt het gemakkelijker om een bundel artikelen tot één document samen te stellen. Hiervoor moet deze class natuurlijk wel het een en ander veranderen in de kernel van \LaTeX , bijvoorbeeld om de commando's `\documentclass` en `\usepackage` en de document omgeving onschadelijk te maken. Ook de verwerking van paginanummers, inhoudsopgave, lijst van figuren e.d. wordt onder handen genomen. Als proef heb ik twee MAPS artikelen geprobeerd te combineren:

```
\documentclass[colclass=maps]{combine}
\title{Combine test}
\begin{document}
\pagestyle{combine}
\maketitle
\tableofcontents
\clearpage

\begin{papers}
\coltoctitle{AucTeX}
\coltocauthor{Piet van Oostrum}
\import{auctex}

\coltoctitle{CTAN Nieuws}
\coltocauthor{P. van Oostrum}
\import{fromctan}
\end{papers}
\end{document}
```

Een korte toelichting: We gebruiken natuurlijk als documentclass **combine** maar we willen formatteren volgens de class **maps**. Dit wordt aangegeven door de optie `colclass=maps`. Bij verstek wordt **article** gebruikt. In

principe moeten de te bundelen artikelen van dezelfde documentclass zijn, maar het is mogelijk met opties aan te geven dat sommige van een andere class zijn. Dit kan echter problemen geven als dezelfde commando's met verschillende definities in die classes gebruikt worden.

De omgeving `papers` wordt als verpakking om de artikelen heengezet. Deze omgeving maakt de commando's zoals `\documentclass` onschadelijk. Het is mogelijk om meerdere `papers` omgevingen op te nemen bijvoorbeeld wanneer verschillende documentclasses ingevoegd moeten worden.

De commando's `\coltoctitle` en `\coltocauthor` geven de ingangen voor de inhoudsopgave van het geheel. Het is mogelijk om met het package **combinet** de gegevens uit de afzonderlijke artikelen te laten halen maar toen ik dit probeerde werkte dit niet vlekkeloos. Het `\import` commando tenslotte voegt het gewenste artikel in.

Er zijn nog verschillende opties en commando's die o.a. te maken hebben met layout, inhoudsopgaven e.d. die ik hier niet zal behandelen. Toen ik dit voorbeeld probeerde kreeg ik een foutmelding op het `\maketitle` commando in het eerste artikel, maar de uitvoer was toch naar verwachting. In het algemeen kan gesteld worden dat het te verwachten is dat \LaTeX lang niet alles foutloos zal verwerken, en ik neem aan dat in de loop van de tijd verschillende problemen opgelost zullen worden.

Een ding wat ik miste was de mogelijkheid een eigen paginalayout te maken met headers/footers e.d. Combine gebruikt nu zijn eigen paginalayout en het is niet mogelijk dit te veranderen. Ik heb geprobeerd een `fancyhdr` layout te gebruiken maar wat ik ook deed, hij ging altijd op de default over. Waarschijnlijk zal wat nauwkeuriger bestudering van de code hiervoor ook wel een mogelijkheid bieden.

Beoordeling

Combine lijkt me een aardig pakket voor dit soort dingen maar ik denk dat men toch al gauw tegen de beperkingen oploopt. Eigenlijk is \TeX gewoon niet ontworpen voor dit soort exercities. Weliswaar heeft \TeX het groeperingsmechanisme waarmee definities lokaal gehouden kunnen worden, maar er zijn omstandigheden dat men *globale* wijzigingen moet maken. Deze zijn dan weer heel erg globaal. Er is niet een soort tussenniveau, en dat zou mijns inziens nodig zijn om dit succesvol te maken.

Wanneer men een macropakket gebruikt men heeft de discipline om hier niet vanaf te wijken dan kan (bij een goed ontworpen pakket) dit wel gedaan worden. Het ontwerp van het pakket moet hier dan al van tevoren rekening mee hebben gehouden. Of het moet tenminste zo opgezet

we een HTML document afdrukken. Ik kon dat met een simpel 3-regelig AWK script oplossen.

zijn dat er niet teveel haken en ogen aan zitten. Bij \LaTeX komt als extra probleem dat er zoveel packages zijn die zich ook nog moeten gedragen. Misschien is dit probleem bij ConTeXt gemakkelijker op te lossen?

Andere bijdragen

Onderstaande bijdragen zijn \LaTeX packages of classes, tenzij anders vermeld. Als men deze wilt ophalen is dit het gemakkelijkst om eerst te zoeken via <http://www.ntg.nl/ctan-search.html>

contour Maak een gekleurde contour om een gegeven tekst

custom-bib Maak je eigen bibstyles zonder coderen (met een vraag/antwoord mechanisme)

webeq Een voorbeeld voor het gebruik van PDF \LaTeX : interactieve documenten

mparhack Een oplossing voor het \LaTeX probleem dat marginpars soms aan de verkeerde kant komen

latexdraw Een tekenprogramma voor \LaTeX pictures voor het X Window system

wmf2eps Een omzetter van WMF naar EPS (MS Windows – shareware)

xtab Een uitbreiding van supertabular

/simplified-latex Een handleiding voor \LaTeX

labelmag Een MS Windows programma om labels te selecteren en te genereren (met \TeX)

lgrind Lgrind is een programma om source code van verschillende programmeertalen naar \LaTeX om te zetten. Er is nu een MS Windows .exe beschikbaar

psnfss Een nieuwe versie van dit pakket om Type 1 fonts in \LaTeX te gebruiken. De source is beschikbaar als psnfss-source

savefnmark Een package om voetnootnummer te kunnen hergebruiken

tff-tetex Een handleiding hoe Truetype fonts in te \TeX te gebruiken

eso-pic Een plaatje op iedere pagina toevoegen

pdfscreen Hiermee kunnen mooie presentaties met pdf \LaTeX gemaakt worden (vergelijkbaar met die van ConTeXt)

titlesec Geeft de mogelijkheid de layout van sectie (en andere) titels te wijzigen

ltx2rtf Omzetter van \LaTeX naar RTF

rtf2latex2e Omzetter van RTF naar \LaTeX

acronym Een package om een lijst van afkortingen te maken (net als een index)

isodate Geeft de mogelijkheid het `\today` commando verschillende formaten (vooral ISO) te laten produceren

epstopdf Programma om EPS files om te zetten naar PDF formaat. gebruikt ghostscript. Er is zowel een Perl als C versie (en Windows .exe)

jurabib Package voor citaten op de (Duitse) juridische manier

tff2pt1 Programma om Truetype fonts om te zetten in Type 1 fonts

varindex Een package om index entries in diverse stijlen te kunnen genereren

expressg Een MetaPost package voor diagrammen

TeXnicCenter Een grafische interface voor TeX op MS Windows systemen. Is nog een beetje buggy

texdoctk Een grafische interface voor te \TeX , gebaseerd op Tk (voornamelijk voor Unix systemen)

mathpazo Fonts met mathematische tekens die gebruikt kunnen worden samen met Palatino

lineno Uitgebreid pakket om automatische regelnummers in de uitvoer te genereren

arydshln Een update van dit pakket om stippelijnen in arrays en tabellen te kunnen gebruiken

KOMA-Script Een aantal documentclasses die als vervanging van de standaard classes gebruikt kunnen worden en die er Europees uitzien

pstoedit Een update van dit programma dat – in samenwerking met ghostscript – Postscript files in een grote variëteit van andere formaten om kan zetten, die dan bewerkt kunnen worden

bibtopic Geeft de mogelijkheid verschillende categorieën bibliografieën op te nemen

ticket Productie van badges, tickets, visitekaartjes e.d. voor conferenties

amsmath Update van dit bekende package

html2text programma om HTML naar platte tekst om te zetten

minutes Package voor het maken van vergaderingsverslagen

ntabbing Tabbing omgeving met regelnummering

chngpage Geeft de mogelijkheid bladspiegel e.d. midden in een document te wijzigen

bbl2html Awk script voor het omzetten van een .bbl file (uitvoer van bibtex) naar HTML

patchcmd Hiermee kan een bestaand L^AT_EX commando uitgebreid worden

cuisine Voor het formatteren van recepten

Tot Slot

Ik zou graag terugkoppeling willen hebben over deze bijdrage: is dit een zinvolle manier om deze informatie door te geven?

ConT_EXt en pdft_Ex

abstract

In deze serie zal het laatst bekende nieuws op het gebied van pdfT_EX en ConT_EXt aan de inner crows worden ontfutseld. In deze eerste aflevering komt aanbod wat de laatste versies van pdfT_EX en ConT_EXt zijn. En tevens waar je ze kunt halen. En verder in iedere aflevering een ConT_EXt tip: vandaag over het produceren van A5 boekjes die op A4 worden afgedrukt.

keywords

Context, pdf(e)-TeX, installatie, nieuws

Inleiding

Hallo ConT_EXt en pdfT_EX gebruikers. En natuurlijk ook de toekomstige gebruikers. Vanaf deze plaats zal ik ieder half jaar de laatste stand van zaken rondom het compleetste macro-pakket en de geavanceerste T_EX-compiler belichten. In deze aflevering zal ik voornamelijk ingaan op wat de laatste versies zijn en waar deze te verkrijgen zijn.

pdfT_EX

Wellicht weet niet iedereen wat pdfT_EX precies is. pdfT_EX is geheel gelijk aan T_EX, maar het kan direct PDF uitvoer produceren. Dit scheelt een aantal handelingen. Wordt namelijk T_EX gebruikt, dan wordt eerst DVI geproduceerd, dan moet dat omgezet worden in Postscript en dat kan vervolgens worden omgezet in PDF.

En voor wie niet weet wat PDF is: PDF staat voor Portable Document Format en is bedacht door Abboe. PDF lijkt veel op DVI, het is namelijk een gecompileerd document. Een voor de gebruiker plezierig verschil is dat de fonts die in het document gebruikt worden in het document zijn opgenomen. Iedereen met een PDF viewer kan het document bekijken zonder meldingen over missende fonts, TFM bestanden en dergelijke. Een T_EX installatie is dus niet nodig om PDF documenten te bekijken. Op <http://www.adobe.com/products/acrobat/adobe.pdf.html> is meer informatie over PDF te vinden.

ConT_EXt heeft graag de laatste pdfT_EX versie. De nieuwste mogelijkheden van pdfT_EX worden vaak het eerst ondersteund in ConT_EXt en omgekeerd worden uitbreidingen en toevoegingen in ConT_EXt soms pas mogelijk met nieuwere versies van pdfT_EX. Een andere belangrijke reden om bij te blijven met pdfT_EX is dat pdfT_EX, anders dan T_EX, actief in ontwikkeling is en daardoor ook fouten bevat die in nieuwere versies opgelost kunnen zijn.

Welke versie van pdfT_EX op een systeem is geïnstalleerd is te zien bij het opstarten van pdfT_EX. Bijv. op mijn systeem:

```
/home/berend# pdftex
This is pdfTeX, Version 3.14159-13d (Web2C 7.3.1)
**
```

Het versienummer van pdfT_EX is opgebouwd uit 3 delen:

- ▣ Het eerste nummer, het getal 3.14159, is het nummer van T_EX zelf. Dit nummer wordt bedacht door Donald Knuth, T_EX's auteur. Dit nummer zal bij iedere nieuwe versie van T_EX een nieuw cijfer van π tonen. Nu wordt T_EX zelf eigenlijk nooit meer gewijzigd aangezien Donald Knuth het systeem bevroren heeft.

□ Het tweede nummer, het getal 13d, is het versienummer van pdf \TeX . Deze versie is al redelijk ouderwets, maar wordt bijvoorbeeld standaard meegeleverd met de 4 \TeX 5.0 distributie. Ook mijn RedHat Linux 6.2 systeem komt met deze versie.

De versie die eigenlijk aanwezig moet zijn is versie 14f, de laatste versie de pdf \TeX 's auteur Han The Thanh heeft goedgekeurd. Hieronder zal ik ingaan op waar deze versie te verkrijgen is.

□ Het laatste nummer, getal 7.3.1, is de versie van het programma dat gebruikt is om de \TeX source code naar C om te zetten. \TeX is geschreven in een soort Pascal. Deze code wordt omgezet in equivalente C code en vervolgens uitgebreid en aangevuld met een aantal patches die van de \TeX de Web2C distributie maken. De Web2C is de \TeX versie die tegenwoordig vrijwel iedereen draait.

Web2C past \TeX zelf niet aan, maar maakt het op zich zeer primitieve \TeX wel geschikter om mee te werken. De interne tabellen van \TeX kunnen bijvoorbeeld hiermee groter worden gemaakt. Ook het zoeken van bestanden is een stuk vriendelijker geworden door het gebruik van de kpathsea library wat in Web2C gebruikt wordt.

Voor de liefhebbers is het wellicht interessant om te weten dat het relatief eenvoudig is \TeX te compileren met Delphi. Het is tenslotte Pascal.

Naast pdf \TeX kennen we ook nog pdf ϵ - \TeX . In deze versie is niet \TeX de basis, maar ϵ - \TeX . In tegenstelling tot Web2C dat \TeX alleen maar omzet in C code, is ϵ - \TeX een uitbreiding op \TeX zelf. Con \TeX t heeft ook ondersteuning voor ϵ - \TeX . Het gebruik van ϵ - \TeX boven \TeX is aan te bevelen omdat Con \TeX t met ϵ - \TeX op sommige punten sneller is.

Mijn pdf ϵ - \TeX versie is:

```
/home/berend# pdftex
This is pdfTeX, Version 3.14159-13d-2.1 (Web2C 7.3.1)
**
```

We zien nu maar liefst 4 versie nummers! De \TeX versie, pdf \TeX versie en Web2C versie zijn gelijk. Getal 2.1 staat voor het versienummer van ϵ - \TeX .

Goed, gezien het versienummer van pdf \TeX wordt het tijd om een nieuwe versienummer te installeren. Con \TeX t klaagt trouwens als de versie van pdf \TeX niet meer bij de tijd is. In de .log file staat dan een melding als:

```
systems      : please update your pdfTeX binaries
```

De startpagina voor pdf \TeX is <http://tug.cs.umb.edu/applications/pdftex>. Binaries voor diverse unix systemen zijn te vinden op <ftp://ftp.muni.cz/pub/tex/local/cstug/thanh/pdftex/0.14f>. Voor mensen die Windows (Windowx 9x of Windows NT) gebruiken zijn er binaries te vinden op bijvoorbeeld <ftp://ftp.dante.de/pub/tex/systems/win32/fptex/pdftex-win32.zip>. Als de binaries uitgepakt zijn, kunnen deze naar de juiste directory gekopieerd worden, bijvoorbeeld /usr/local/bin (BSD) of /usr/bin (Linux) of c:/4TEX5.0/BIN/win32 (4 \TeX). Zorg dat ook de .pool files op de juiste plek komen, bijvoorbeeld /usr/share/texmf/web2c.

Als laatste link mag de pdf \TeX FAQ niet ontbreken: <http://tug.cs.umb.edu/applications/pdftex/pdfTeX-FAQ.pdf>. Onder andere komen aan bod fonts en graphics.

pdf \TeX nieuws

Hoewel ik in de vorige paragraaf vertelde dat pdf \TeX gewoon \TeX is, maar dan met PDF uitvoer in plaats van DVI uitvoer, is dat gelukkig niet helemaal waar. In pdf \TeX zitten enkele mogelijkheden die documenten er nog prachtiger uit laten zien. Bekijk de voorbeelden maar eens op <http://www.pragma-ade.nl/samples.htm>. Let op de uiterst strakke rechterkantlijn, het gevolg van het aanzetten van de optie 'margin kerning'.

De auteur van pdfTeX, Han The Thanh, heeft bijna een dissertatie afgerond over pdfTeX en allerlei algoritmes voor het verfraaien van de opmaak van een document. Ik verwacht in de volgende MAPS de link naar zijn dissertatie te kunnen geven.

ConTeXt

Als er nieuwe pdfTeX binaries zijn geïnstalleerd, moeten de ConTeXt format files opnieuw aangemaakt worden, één voor elke taal die ConTeXt ondersteunt. Dit kan eenvoudig met het volgende commando:

```
texexec --make
```

Hiermee wordt het format voor een aantal interfaces opnieuw gemaakt. Om de format voor een specifieke interface opnieuw te maken, kan het volgende commando gebruikt worden:

```
texexec --make --interface=en
```

Als de foutmelding

```
fmtutil: no info for format 'cont-nl'.
```

optreedt, dan moet het configuratiebestand `fmtutil.cnf` aangepast worden. Dit kan met:

```
fmtutil --edit
```

Als er een regel staat als:

```
# cont-nl      pdfetex      cont-usr.tex   *cont-nl.ini
```

zorg dan dat het commentaar, het hekje verwijderd wordt, en vervang `.ini` door `.tex`, zodat er komt te staan:

```
cont-nl      pdfetex      cont-usr.tex   *cont-nl.tex
```

Bestaat zo'n regel niet, voeg deze dan toe. Als in de laatste kolom niet `cont-nl.tex` staat maar `cont-nl.in` dan zal de foutmelding zijn:

```
! I can't find file 'cont-nl.ini'.
```

Ook als er een nieuwe ConTeXt distributie opgehaalt wordt, zullen de format files opnieuw moeten worden aangemaakt. De laatste ConTeXt interfaces zijn te verkrijgen via <http://www.pragma-ade.com/pragma-ade/download.htm>. Het valt wellicht niet op, maar deze URL eindigt op `.com` in plaats op `.nl`! De nederlandse ConTeXt site heeft nogal ruimte gebrek, op deze Amerikaanse site van Pragma is nog ruimte genoeg en zal waarschijnlijk ook meer materiaal gaan bevatten. Ik raad dan ook iedereen aan om zijn link naar de homepage van Pragma te wijzigen in <http://www.pragma-ade.com>.

De laatste stabiele distributie op moment van schrijven van dit artikel (uiteraard vlak voor de deadline van 1 augustus) is die van 29 mei 2000. De versie van ConTeXt op een bepaald systeem is te achterhalen met:

```
context &cont-nl
```

Een van de eerste regels die dan getoond wordt is het versienummer:

```
ConTeXt ver: 2000.5.29  fmt: 2000.7.29  int: dutch  mes: dutch
```

Na het versienummer volgt de datum waarop voor het laatst de format file opnieuw is gemaakt.

Bij het vervangen van de ConTeXt files is het *uitermate, uitermate en nog eens uitermate* belangrijk om de vorige versie goed te verwijderen! Wel 90% van fouten bij het installeren van nieuwe versies komen door het niet goed verwijderen van een vorige versie. De stappen die uitgevoerd moeten worden:

```
[n][uitlijnen=normaal]
```

- Verwijder de `tex/context/base` directory, of beter nog, move deze naar een veilige plek. Deze directory is te vinden onder `/usr/local/share/texmf` (BSD) of `/usr/share/texmf` (Linux) of voor Windows gebruikers op `c:/4TEX5.0/TEXMF`.
- Pak de nieuwe ConT_EXt distributie uit in de bovengenoemde `texmf` directory.
- Zorg dat alle bestanden vindbaar zijn door `mktexhash` of `mktexlsr` te draaien.
- Update dan `texexec` en `texutil`. Deze bestanden staan na uitpakken in de bovengenoemde directory in `context/perl`.

Onder Linux is het een aardige truuk om het bestand `/usr/bin/texutil` te linken naar de directory waar default de ConT_EXt `texutil.pl` staat:

```
ln -fs \
  /usr/share/texmf/context/perl/texutil.pl \
  /usr/bin/texutil
```

Ditzelfde kan gedaan worden voor `texexec`. Onder Windows is het het beste om de nieuwe bestanden over de bestaande te kopiëren.

De laatste versies ten tijde van schrijven van dit artikel zijn:

- `texexec`: versie 2.1
- `texutil`: versie 7.3
- Run `texexec -make` om de formats opnieuw aan te maken.

Beta versies van ConT_EXt zijn te vinden op <http://www.pragma-ade.com/pragma-ade/beta.htm>. Nieuwe features zijn eerst beschikbaar in de beta versies, dus in deze pdfT_EX en ConT_EXt nieuwsrubriek zullen we meestal beta's nodig hebben. Het maken van format files voor beta's gaat uiteraard op dezelfde manier als hierboven beschreven..

ConT_EXt nieuws

Na al deze voorbereidingen wordt het tijd om eens te kijken naar recente ConT_EXt features. Het command `\stelpaginnummer` is recent uitgebreid. De optie start kan nu worden gezien van de waarden:

- `start`: toon het paginanummer en verhoog het paginanummer.
- `stop`: toon het paginanummer niet, en verhoog het niet.
- `handhaaf`: toon het paginanummer, maar verhoog het niet. Deze optie leek bij mij niet te werken, maar functioneerde als een synoniem voor `stop`.
- `leeg`: toon het paginanummer op *deze* pagina niet, maar verhoog het nummer wel. Op de volgende pagina wordt het nummer wel weer getoond.
- `geen`: toon het paginanummer niet meer, maar verhoog het nummer wel.

Een voorbeeld:

```
% interface=nl
\stelpaginnummerin
  [status=start]
\starttekst
\hoofdstuk{Eerste}
Dit is pagina 1.
\pagina
En dit twee.
\pagina
```

```

\stelpaginanummerin
  [status=stop]
Pagina 3 krijgt geen nummer.

\pagina

\stelpaginanummerin
  [status=start]
Dit is pagina 4,
maar is genummerd als pagina 3.

\pagina

\stelpaginanummerin
  [status=geen]
Deze pagina krijgt paginanummer 4,
maar het paginanummer wordt niet getoond.

\pagina

\stelpaginanummerin
  [status=start]
En dit is dan paginanummer 5.

\stoptekst

```

De volgende mogelijkheden zijn op dit moment nog in beta, dus niet beschikbaar in de officiële release.

Na de nieuwe `\starttabulatie` is er nu ook `\bTABLE ... \eTABLE`. Dit is een nieuwe manier van het zetten van tabellen. Moest er met de oude methode exact gespecificeerd worden hoe het gewenste resultaat bereikt moest worden —zoveel kolommen, zo breed, e.d.— met de nieuwe methode hoeft slechts de invoer vast gelegd te worden, het aantal kolommen en de breedte worden automatisch berekend. Met deze nieuwe tabelfunctionaliteit is het gemakkelijker geworden om op een bepaalde regel kolommen samen te voegen of in een bepaalde kolom bepaalde rijen samen te voegen. Achtergronden en kleur zijn uiteraard prominent aanwezig.

De nieuwe mogelijkheden zijn beschreven in ConT_EXt up-to-date 2000/8, zie de ConT_EXt Up-To-Date pagina op <http://www.pragma-ade.com/pragma-ade/uptodate.htm>.

Voor mensen die met de engelse interface van ConT_EXt werken is er nu een behoorlijk complete handleiding beschikbaar, zie <http://www.pragma-ade.com/pragma-ade/manual.htm>.

De ConT_EXt tip

Laat ik met dit eerste artikel direct maar een traditie beginnen om af te sluiten met een ConT_EXt tip. ConT_EXt zou ConT_EXt niet zijn als moeilijke dingen niet relatief makkelijk zouden zijn. Het volgende voorbeeld toont hoe je, met enkele simpele opdrachten in je initialisatie-sectie, een A5 document kunt typesetten op A4 papier. Dit is dus geen verkleintruuk, je document wordt, in de ingestelde fontgrootte, getypeset op A5. Twee A5 pagina's worden op een A4 geplaatst. Het geheel wordt geroteerd zodat het direct vanuit Acrobat afgedrukt kan worden.

```

% interface=nl
% vanaf hier
\stellayoutin

```

```
[breedte=midden,hoogte=midden]
\stelpapierformaatin
[A5][liggend,A4]
\stelarrangerenin
[2SIDE,geroteerd]
% tot hier

\starttekst

\hoofdstuk{Eerste}
Dit is pagina 1.

\pagina

En dit twee.

\hoofdstuk{Twee}

En dit is pas de tweede fysieke pagina!

\stoptekst
```

Dat was het wat mij betreft voor deze keer. En eens kijken, oef, vijf voor twaalf, toch de deadline gehaald!

Gezeefd uit de TEX-NL discussielijst

abstract

This is part seven in a series of articles started in 1993 by Philippe Vanoverbeke. Philippe selected a number of messages from the TEX-NL mailing list which deserved to be remembered and looked up by readers of MAPS. Solutions, hints and tips. MAPS of fall 1999 had part 5. This installment more or less covers TEX-NL august 1999 through july 2000. During a hectic year I had missed most messages and working through the backlog for this article provided me with a useful and time consuming opportunity to see what I would have missed otherwise. Questions and answers are marked in twins but sometimes a reply poses a new question, invoking one more answer, making for an instructive thread. As before, honorary NTG member Piet van Oostrum, and also Hans Hagen are always helpful members of the list. To them and all the others feeding the list with their stimulating questions and helpful answers thanks!

Edwin Drost

Tabellen verdelen

Ik wil een tabel met drie kolommen precies even breed hebben als de tekstbreedte. Wie weet hoe ik een deling kan uitvoeren, zodat ik iets krijg als.

```
\begin{tabular}{| p{\textwidth / 2} | p{\textwidth / 4}
               | p{\textwidth / 4} |}
...
\end{tabular}
```

A.W.J. van der Meer

Tabellen verdelen Dat gaat niet zonder meer goed: je moet ook nog rekening houden met het feit dat er ruimte tussen de kolommen zit. Ik heb het wel eens zo gedaan:

```
\newlength{\ceen} \setlength{\ceen}{0.5\textwidth}
\addtolength{\ceen}{-2\tabcolsep}
\newlength{\ctwee} \setlength{\ctwee}{0.25\textwidth}
\addtolength{\ctwee}{-2\tabcolsep}
\newlength{\cdrie} \setlength{\cdrie}{0.25\textwidth}
\addtolength{\cdrie}{-2\tabcolsep}
```

... en dan kun je \ceen enz. gebruiken als kolombreedtes.

Wybo Dekker

Wat denk je hiervan:

```
\documentclass[12pt]{article}
\usepackage{tabularx}
\parindent0pt
\textwidth100mm
\begin{document}
Hier is een tekst om de paginabreedte te laten zien, een flink lange regel
Hier is een tekst om de paginabreedte te laten zien, een flink lange regel
Hier is een tekst om de paginabreedte te laten zien, een flink lange regel
\\
\begin{tabularx}{\textwidth}{|>\hspace=.5\hspace}X
                    |>\hspace=.25\hspace}X
```

```

|>{\hsize=.25\hsize}X|}
kolom 1&kolom 2&kolom 3\\
kolom 1&kolom 2&kolom 3\\
kolom 1&kolom 2&kolom 3\\
\end{tabularx}
\end{document}

```

Hans Hagen In Context:

```

\begin{tabular}
\starttabulatie[|p|p|p]
\NC whatever text \NC whatever text \NC whatever text \NC\NR
\stoptabulatie
\end{tabular}

```

also breaks across pages.

Piet van Oostrum Wybo Dekker schrijft:

```

\begin{tabularx}{\textwidth}{|>{\hsize=.5\hsize}X
|>{\hsize=.25\hsize}X
|>{\hsize=.25\hsize}X|}

```

Volgens de documentatie zouden de coëfficiënten van `\hsize` moeten totaliseren tot het aantal X'en. Dan zou het dus moeten zijn: 1.5 resp .75.

Dat het toch werkt komt waarschijnlijk doordat er alleen X kolommen zijn en geen andere.

Sven Bovin **ConTeXt handleiding**

Een vraagje voor de PDF-wizards:

Gisteren heb ik van de NTG-server de ConTeXt-handleidingen (de on screen en de papierversie) gedownload en (de papierversie) afgeprint. En daar zit ik met een probleem: pagina 276 (280 absoluut) wil op de PostScript-printer niet afdrucken, tenzij ik haar als afbeelding verstuur. Dat laatste resulteert in bedroevende kwaliteit en een geroteerde pagina. De inktjetprinter heeft er geen problemen mee, maar roteert ook de pagina. Afdrucken naar een PostScriptbestand geeft ook problemen: Ghostscript geeft een foutmelding en stopt er mee. Heeft iemand hier een oplossing voor?

Taco Hoekwater Bij mij print ie prima (hp4000), maar dat is dan ook een level2 postscript printer. Probeer het eens met de 'ps level 1' print setting. Gs5.5 is ook perfect gelukkig met dit bestand.

Sven Bovin Wij hebben hier een HP 5M, en ik dacht dat dat toch een level 2-printer is. Maar als level 1-PostScript printen helpt inderdaad. De pagina komt er nu mooi uit. Bedankt voor de tip.

Hans Hagen Aha. Bij printen wordt door de viewers een soort pdf naar de printer gestuurd voorafgegaan door een pdf interpreter (geschreven in ps). Context neemt plaatjes zodanig dat ze slechts eenmaal worden opgenomen, ook als meerdere malen gebruikt, (object reuse). In ps2 kan dat redelijk (xforms), maar kost een bulk geheugen en is niet helemaal betrouwbaar (ps level twee misfeature); bij printen in level 1 worden geen PS form objecten gebruikt, en dat is wel zo veilig.

Rob Rutten **Verbatim**

Ik wil graag in een eigen stylefiletje verbatim zetten tussen |. Dus: |dit is verbatim| en dit niet.

Ruud de Rooij `\RequirePackage{shortvrb}`
`\MakeShortVerb{\}`

Piet van Oostrum fancyvrb heeft hiervoor een commando:
`\DefineShortVerb{\}`

Komt ook in docstrip voor.

Taco Hoekwater Of in context met supp-ver:
`\input supp-ver`
`\catcode\|= \active \def|{\type|}`

*Erik Frambach***emlines**

Ik heb een stel oude L^AT_EX-bestanden waarin met emlines (van emTeX, ook door dvips ondersteund) grafieken zijn gemaakt. Op moderne printers komen die lijnen echter te dun uit. emline is als volgt gedefinieerd:

```
\def\emline#1#2#3#4#5#6{%
  \put(#1,#2){\special{em:moveto}}%
  \put(#4,#5){\special{em:lineto}}}
```

Hoe kan ik die lijnen dikker maken, liefst vanuit de T_EX-file?

Piet van Oostrum

Probeer eens `\special{em:linewidth 2pt}` of een andere waarde.

*Wybo Dekker***lijstjes in tabel**

Ik wil in een tabel een aantal lijstjes kwijt, een soort itemize maar met normale regelafstand en zonder inspringen. Ik wil dus:

```
-----
123  * tekst
      * andere lange-
      re tekst
-----
124  * tekst
      * andere lange-
      re tekst
-----
```

Ik probeerde (Kopka p. 206) het onderstaande, maar krijg boven iedere lijst een flinke lege ruimte, zodat “123” en “* tekst” niet meer op 1 regel staan. Ik heb met alle lijstlengtes geëxperimenteerd maar krijg het niet voor elkaar. Wie helpt?

```
\documentclass{book}
\usepackage{array}
\parindent0pt
\newcolumnntype{R}[1]{>{\raggedright}p{#1mm}}
\newenvironment{lijst}[1]{%
  \begin{list}{}{%
    \settowidth{\labelwidth}{#1}
    \setlength{\leftmargin}{\labelwidth}
    \addtolength{\leftmargin}{\labelsep}
    \setlength{\itemsep}{0ex}
    \setlength{\parsep}{0ex}
    \renewcommand{\makelabel}[1]{##1\hfill}}
```

```

    }
  }
{\end{list}}
\begin{document}
\noindent
eerste regel\
tweede regel

\begin{tabular}{@{}rR{25}}\hline
123
&
\begin{lijst}{x}
\item[p] tekst
\item[q] tekst
\end{lijst}\tabularnewline\hline
123
&
\begin{lijst}{x}
\item[p] tekst
\item[q] tekst
\end{lijst}\tabularnewline\hline
\end{tabular}
\end{document}

```

Taco Hoekwater Gewoon zelf doen:

```

\newdimen\leftside
\newenvironment{lijst}[1]{%
  \settowidth{\leftside}{#1}%
  \addtolength{\leftside}{\labelsep}%
  \let\item\myitem}\endgraf\kern -10pt}
\def\myitem[#1]{%
  \endgraf
  \parindent \leftside
  \hangindent \leftside
  \indent \llap{\hbox to \leftside{#1\hss}}%
  \ignorespaces }

```

Sven Bovin **Courier als monospaced font**

Ik heb hier een document waarin ik Courier wil gebruiken als monospaced font, met Computer Modern als basisfont (gewoon om te kijken of Courier beter combineert met cmr dan cmrtt, die ik persoonlijk te vet vind voor cmr). Hoe leg ik dat aan boord? Het package 'courier' doet het niet. Ik gebruik namelijk definities als

```
\newcommand{\Bestand}[1]{\texttt{\itshape{#1}}}
```

om bestandsnamen op te maken, en dan krijg ik met dat package

- a. geen monospaced naam
- b. een omgekeerd uitroepteken *voor* en een omgekeerd vraagteken *na* de bestandsnaam

wanneer ik `\Bestand{foo}` gebruik.

Wybo Dekker **raar afbreekprobleem**

In het onderstaande voorbeeld staan twee identieke regels met achter de eerste een lange superscript. Die superscript past niet meer op de regel, maar er wordt niet afgebroken, blijkbaar wordt `admissions.3,4,6-48` niet als afbreekbaar herkend. In de tweede zin, waar `admissions` ook te lang voor de regel is, wordt netjes tussen de twee s-en afgebroken. Dat zou in de eerste regel dus ook mooi hebben gekund.

Ik heb van alles geprobeerd ertussen te zetten: `{}` of `\mbox{}` of `\rule{0pt}{0pt}`. Niets helpt, behalve `\hspace{0pt}`. Alleen: dan is de superscript afbreekbaar geworden zodat die in de derde zin op z'n eentje op een nieuwe regel terecht komt.

Nu hoop ik maar dat iemand een beter idee heeft...

```
\documentclass[12pt]{book}
\usepackage{a4,times}
\newcommand{\SUP}[1]{%\hspace{0pt}%
                    \textsuperscript{#1}}

\begin{document}
\chapter{Inleiding}
The increase in the incidence of heart failure has led to
an increase in hospital admissions.\SUP{3,4,6--48}
The increase in the incidence of heart failure has led to
an increase in hospital admissions.\\
The increase in the incidence of heart failure has led to
an increase in increase hospital.\SUP{123}
\end{document}
```

Taco Hoekwater `\newcommand{\SUP}[1]{\penalty 10000
 #1}`

Lex Molenaar **URL**

Na een paar uur zoeken heb ik nog steeds geen LaTeX Package gevonden waarmee ik web-sites a la `http://www.homepage.nl` in mijn lopende tekst kan opnemen of als Bib cite.

Hyperref is er voor interactieve documenten, en html pagina's maken is ook niet mijn doel. Kortom, als ik een softwarepakket, instelling of bedrijf noem in mijn dissertatie, wil ik graag refereren aan een www adres.

Sven Bovin Als ik mij niet vergis kan het package `url` dat (maar zeker ben ik niet: ik heb het package nog nooit gebruikt).

Geert Stremersch **Blokmatrices**

Ik werk met blokmatrices waarin ik graag horizontale en verticale lijnen zou hebben die echter niet de over de volle breedte of hoogte worden doorgetrokken, bv.

```
( x | x | x )
(-----| )
( x | x | x )
(-----)
( x x | x )
```

Bestaat daar een geschikte package voor LaTeX? Ik heb `multirow` en `multicolumn` bekeken in de Companion, maar kom niet tot een oplossing.

Ik zou ook graag hebben dat (bv binnen een equation environment) die matrix voorafgegaan kan worden door een andere formule, bv. $f(x) = [matrix].g(x)$ Hoe doe ik dit?

Johan Wevers Het volgende doet wat jij wilt:

```
\def\longmid{\rule[-4pt]{0.4pt}{14pt}}
\[
\left(\begin{array}{cp{1pt}cp{1pt}c}
X & \longmid & X & \longmid & X \\
\cline{1-3}
X & \longmid & X & \longmid & X \\
\hline
X & & X & \longmid & X
\end{array}\right)
\]
```

P.Ebbes@research-int.com

Inspringen in enumerate

Ik heb een vraagje over het inspringen van de enumerate omgeving. Ik probeer al een tijdje uit te vogelen hoe of de inspring verkleind kan worden. Dit zal wel iets zijn met renewenvironment ofzo, maar ik kan niet vinden hoe precies. Kunnen jullie mij hierbij helpen? Verder, in welk bestandje staan de officiële definities die L^AT_EX gebruikt, dus in welke file wordt bijvoorbeeld de ‘enumerate-omgeving’ gedefinieerd?

Wybo Dekker Hier is een voorbeeld:

```
\documentclass{article}
\parindent0pt
\begin{document}
We zetten de leftmargin voor 1 niveau diep geneste itemizes
(dus \verb|\leftmargin|) op de breedte van de markers die
we willen gaan gebruiken (hier --) en tellen er \verb|\labelsep|
bij op zodat de itemmarkers tegen de linker kantlijn komen:
\settowidth{\leftmargin}{--}
\addtolength{\leftmargin}{\labelsep}
\begin{itemize}\itemsep-lex
\item[--] een item
\item[--] een item
\item[--] een item
\item[--] een item
\end{itemize}
Bovendien hebben we met een negatieve \verb|\itemsep| de
items dichter op elkaar gezet.
\end{document}
```

Sven Bovin

Font voor tabel kleiner

Hoe kan ik een L^AT_EX de tabellen in een kleiner font zetten dan de broodtekst? Het gaat om een document in de standaard article class. Ik heb geprobeerd met

```
...
{\small{
\begin{table}
...
}}
```

```

\end{table}
}}
...

```

maar dat werkt blijkbaar niet.

Piet van Oostrum

```
\begin{table}
\small ....
\end{table}
```

Maarten Wisse **Kopteksten met boektitel en hoofdstuk**

Een L^AT_EX-vraag. Ik ben `book.cls` aan het aanpassen om een soort klassieke Cambridge-look te krijgen. Daar hoort bij dat ik de kopteksten aanpas, zodat aan de linkerkant de titel van het boek, en aan de rechterkant de hoofdstuktitel te zien is.

Het tweede heb ik gedaan door `\leftmark` te zetten waar `\rightmark` stond. Het eerste probeerde ik te doen door `\title` te zetten waar `\leftmark` stond. Dat werkt niet. Waarom niet en wat moet ik wel doen?

Piet van Oostrum Ten eerste is `\title` niet een ‘variabele’ die de titel zet, maar een commando om de titel te definiëren. Die wordt dan in een andere ‘variabele’ `\@title` gezet, maar die wordt om geheugen te besparen weer leeggemaakt bij de `\maketitle`. Als je dus toch een eigen class file maakt sloop die leegmaking er dan uit (`\global\let\@title\@empty`), of bewaar de zaak nog extra in een eigen commando.

Edwin Drost **Opvullen tabel**

Ik wil een tabel met 2 kolommen maken:

```

\begin{tabular}{|l|p{4.5cm}|}
  bla1 & bla2 \\
...
\end{tabular}

```

Nu wil ik eigenlijk dat de breedte van de tweede kolom automatisch wordt bepaald, zodat de volledige breedte gelijk is aan `\textwidth`. Wie weet of dit mogelijk is en zo ja hoe?

Piet van Oostrum

```
\usepackage{tabularx}
...
\begin{tabularx}{|l|X|}
  bla1 & bla2 \\
...
\end{tabularx}
```

Rob Rutten **Glossary style voor article class**

Bestaat er een package dat het volgende voor mij doet:

```

\begin{glossary}
  \item{Aap} Wezen dat verdacht veel gelijkenissen vertoont met...
  \item{Mens} Wezen dat verdacht veel gelijkenissen vertoont met...
\end{glossary}

```

Waarbij de output als volgt gezet wordt:

```

{\bf Aap:}\quad Wezen dat verdacht veel
gelijkenissen vertoont met...
{\bf Mens:}\quad Wezen dat verdacht veel
gelijkenissen vertoont met...

```

- Werenfried Spit* De standaard L^AT_EX-omgeving `description` doet dat. In de L^AT_EX companion staan wat voorbeelden om dat fine te tunen.
- Piet van Oostrum* Of de outout op **die** manier gezet wordt weet ik niet, maar is een `gloss` package.
`ftp://ftp.ntg.nl/pub/tex-archive/macros/latex/contrib/supported/gloss`
- Rob Rutten* Werenfried Spit schreef ‘De standaard L^AT_EX-omgeving `description` doet dat, maar dan zijn de stukjes verschillend uitgelijnd.
 De companion heb ik niet bij de hand. Piet van Oostrum noemde het `gloss` package. Tja, dat package is in het spaans (of portugees?) en dat snap ik niet. Ik heb trouwens zelf even het volgende gedefinieerd:
- ```
\begin{verbtim}
\def\g#1{\goodbreak\vskip0pt plus .5pt\par
\hangindent .5cm\noindent{\bf #1: }\ignorespaces}
```
- en met:
- ```
\g{Aap} Een wezen etc.
\g{Mens} Een wezen ...
```
- Gaat het prima.
-
- Eric Van Steenberg* **Math symbolen**
 Voor de wiskundige knobbels onder u heb ik een klein vraagje. Ik ben bezig een boek te zetten in L^AT_EX maar kan enkele symbolen niet vinden. Het gaat om volgende symbolen:
- ▣ het spiegelbeeld van `\models` (zoals `\dashv` het spiegelbeeld is van `\vdash`)
 - ▣ de negatie van `\models` (zoals `\nvDash` de negatie is van `\vDash`; ik gebruik nu deze twee omdat die sterk op `models` lijken).
 - ▣ de negatie van `\approx`
 - ▣ drie keer het `\sim` teken boven elkaar
 - ▣ een rechthoekige driehoek met de rechte hoek rechtsonder
- Bestaat er een package waar deze symbolen in staan? Zo ja, graag vermelding waar ik dit kan vinden.
- Geert Stremersch* Negatie kun je denk ik altijd krijgen door `\not` voor het symbool te plaatsen bv. `\not\approx` en `\not\models`
- Piet van Oostrum* Geen package (en vooral: geen font) dat ik weet. Misschien kun je zelf knutselen:
- ▣ het spiegelbeeld van `\models` met een = en een |
 - ▣ drie keer het `\sim` teken boven elkaar gewoon door ze in een boxje te zetten (Knuth maakt sommige symbolen zelf ook zo).
 - ▣ Een rechthoekige driehoek door een schuin lijnje uit het `line10` font te nemen en met `rules` aan te vullen.
- Of natuurlijk zelf metafont te gebruiken.
- Taco Hoekwater* De fonts waar ik mee bezig ben hebben of krijgen al deze tekens (zeker nu er iemand om gevraagd heeft), maar daar schiet je nu niet zo heel veel mee op, ben ik bang. Bovendien zijn die ontworpen voor gebruik bij Times. Aan de andere kant, kijk eens naar
<http://www.cybercomm.nl/~bitttext/fonts.html>

*Eric Van Steenberg***Lijnen naast paragraaf**

In een boek moeten een paar paragrafen in het oog springen. De auteur wenst dit te doen door die paragrafen in te laten springen in combinatie met een verticale lijn links naast en over de volledige hoogte van die paragrafen. In een quotation environment bijvoorbeeld kan ik die tekst laten inspringen maar hoe kan ik daar die lijn nu naast krijgen? Bestaat er een speciaal environment waarin dit kan?

Piet van Oostrum

Als de par afgebroken moet worden aan het eind van de pagina, terwijl de lijn dan moet doorlopen op de volgende pagina, dan wordt het moeilijk. Het package changebar zou hiervoor moeten kunnen zorgen, maar in de praktijk zitten daar nogal haken en ogen aan.

Als het een korte par betreft die altijd op dezelfde pagina blijven dan is het simpel te doen door ze in een tabular te zetten (of nog beter tabularx) met de streep verzord door de tabular:

```
\begin{tabularx}{|X}
```

Taco Hoekwater

Het is nogal tricky ja. Hier is een stukje code dat ik zelf met succes heb gebruikt (geen idee of het beter of slechter dan changebar is, waarschijnlijk eerder slechter...)

```
\begin{barred}
.....
\end{barred}

\newdimen\maxheight
\def\barred{\par\beginngroup
  \global\maxheight =\pagegoal
  \ifdim \maxheight > 100in \maxheight = \vsize \fi
  \global\advance\maxheight -\pagetotal
  \ifdim \maxheight < 36pt \newpage \global\maxheight\vsize \fi
  \global\advance\maxheight-12pt
  \addvspace{12pt}%
  \global\setbox2=\vbox \bgroup
  \advance\hsize -12pt }

\def\barredbox#1{\vskip 0pt \hbox to \hsize{%
  \vrule height \ht#1 depth \dp#1 width .5mm \hss \box#1}}

\def\endbarred{\par\egroup
  \endgroup\par
  \dimen2=\ht2
  \ifdim \dimen2 < \maxheight \barredbox2
  \else \splittopskip=.5em
    \setbox4=\vsplit2 to \maxheight
    \barredbox4
    \newpage
    \loop \ifdim \ht2 > \vsize
      \setbox4=\vsplit2 to \vsize
      \barredbox4
      \newpage
    \repeat
  \barredbox2
  \fi
  \addvspace{12pt}%
}
```

C.C.M. Moes Ik heb het als volgt gedaan:

```
\long\def\MetLijn#1{{\parindent0pt\par%
\hbox{\hskip-1em\vrule\hskip1em\ vbox{#1}}}}
```

Als je de vormgeving iets anders wilt, is dat gemakkelijk aan te passen.

Jozef Pijnenburg **begin end verbatim**
Ik wil graag een afkorting definiëren voor het begin en end van een environment. Bijvoorbeeld:

```
\newcommand{\be}{\begin{\verbatim}}
\newcommand{\ee}{\end{\verbatim}}
```

Zodat ik in het L^AT_EX document i.p.v. `\begin{verbatim}... \end{verbatim}` voortaan `\be` respectievelijk `\ee` kan gebruiken. Deze ‘methode’ levert echter wel een foutmelding (en geen resultaat) op.

Piet van Oostrum Ja, de verbatim omgeving kun je niet in een ander commando gebruiken. De definitie van `\be` is op zich geen probleem. `\ee` wel en dat is ook logisch, want als je in een verbatim `\ee` zet dan wordt er gewoon `\ee` afgedrukt. Dat is de betekenis van verbatim. Het enige dat de verbatim eindigt is de letterlijke tekst `\end{\verbatim}`.

Het zou natuurlijk kunnen als je een verbatim commando had die de afsluitende tekst als parameter meekrijgt, en dat is ook niet zo vreselijk moeilijk (al zijn er wel haken en ogen i.v.m. de catcodes), maar voorzover ik weet is die er niet, en zal je het dus zelf moeten definiëren.

De packages `verbatim` en `fancyvrb` doen allerlei interessante dingen met verbatim maar niet deze. Maar ze zouden als inspiratiebron kunnen dienen.

Taco Hoekwater mbv ConTeXt en/of de `supp-ver.tex` macros voor L^AT_EX kan dit wel:

```
\def\be {\bigskip \processdisplayverbatim{\ee}}
\def\ee {\bigskip}
```

Lex Sijtsma **Watermerk**
Kan iemand mij vertellen of het mogelijk is om watermerken toe te voegen aan pagina's in L^AT_EX? Zodat ik bijv. de tekst ‘Concept’ achter de lopende tekst kan zetten. Kan ik dit doen met een plaatje? Anders?

Sven Bovin Package `draftcopy` (mogelijk moet daar wat in gehackt worden om de juiste tekst te krijgen).

Ben Geels **uitbreiding exercise.sty**
Ik gebruik voor opgaven `exercise.sty`. Daarmee worden antwoorden verzameld en elders, met het commando `\listanswers` gezet. Nu zou ik deze style willen uitbreiden zodat niet alleen antwoorden verzameld worden maar ook, apart uitwerkingen, zoiets:

```
\begin{exercise}
bereken 2(3+4)
\uitwerking
eerst 3+4=7 en daarna 2x7=14
\answer
14
\end{exercise}
```

En dan:


```
\section{Antwoorden}\listanswers
\section{Uitwerkingen}\listuitwerkingen
```

`exercise.sty` schrijft echter alles tussen `\answer` en `\end{exercise}` verbatim weg in een .ans bestand. Daar gaat het dan verder mis met de uitwerkingen, tenminste zoals ik dat geprobeerd heb. Wie ziet hierin een uitdaging?

Hans Hagen Vijf jaar geleden zag ik dat als een uitdaging, vandaar:

```
\definieerblok[vraag]
\definieerblok[antwoord]

\verbergblok[antwoord]

\hoofdstuk{vragen}

\beginvanvraag
  de vraag
\eindvanvraag
\beginvanantwoord
  het antwoord
\eindvanantwoord

enz enz

\hoofdstuk{antwoorden}

\gebruikblokken[vraag,antwoord]
```

Men kan zoveel soorten blokken definiëren als wenselijk, blokken indelen in subgroepen, nummers handhaven danwel opnieuw nummeren, enzovoorts. Binnen zo'n blok kan men enumeraties gebruiken, om vragen en antwoorden te nummeren, die kan men weer aan elkaar koppelen, in geval van interactieve documenten enzovoorts. Ik heb net wat gerommeld met een stijltje voor interactieve examens, maar dat terzijde.

Het slechte nieuws is dat het niet in L^AT_EX werkt, maar standaard context functionaliteit is.

M.Rupper **Geen figuur-referenties in tekst**

Als ik het onderstaande stukje L^AT_EX compileer (2x), staan er geen getallen bij de referenties naar 'rechter fig' en naar 'linker fig'. Hoe los ik het op? (Als ik naar de pagina waarop het staat refereer doet'ie het wel goed...). De tex-code:

```
Hier staat allemaal tekst... Hier staat allemaal tekst... hier en hier
\begin{figure}[tbh]
\setlength{\unitlength}{1cm}
\begin{minipage}[t]{3cm}

  \begin{picture}(3.0,2.5)

    de eerste tekst...

  \end{picture}\par
  \label{linker fig}
  \caption{Links}
\end{minipage}
\hfill
\begin{minipage}[t]{4cm}

  \begin{picture}(4,3)
```

```

        de tweede tekst ..
    \end{picture}\par
    \label{rechter fig}
    \caption{Rechts}

\end{minipage}
\end{figure}

Links staat figuur
\ref{linker fig}
en rechts figuur
\ref{rechter fig}

```

Bert Frederiks Je moet die referentie, dacht ik, in ieder geval achter de `\caption` zetten, omdat je anders het paginanummer als referentie krijgt in plaats van het nummer van het plaatje (als ik me niet vergis, want ik doe nooit anders). Maar jij krijgt nog iets anders te zien, blijkt. Zou dat niet door de spatie komen? Ik doe het altijd zo, en dan werkt het bij mij:

```

        \caption{Rechts} \label{fig:rechter}
\end{minipage}
\end{figure}
    Links staat figuur
    \ref{fig:linker} en
    rechts figuur \ref{fig:rechter}

```

Wybo Dekker **Oplossing:**

```

\documentclass{article}
\begin{document}
\begin{figure}[tbh]
\setlength{\unitlength}{1cm}
\begin{minipage}[t]{3cm}

    \begin{picture}(3.0,2.5)

        de eerste tekst...

    \end{picture}\par
    \caption{\label{linker fig} Links}
\end{minipage}
\hfill
\begin{minipage}[t]{4cm}

    \begin{picture}(4,3)

        de tweede tekst ..

    \end{picture}\par
    \caption{\label{rechter fig} Rechts}

\end{minipage}
\end{figure}

Links staat figuur \ref{linker fig} en rechts figuur \ref{rechter fig}

```

PS. Het is handig (en leidt tot een sneller antwoord) als je bij een probleemvoorbeeld meteen de `\documentclass{}` en `\begin{document}` en dergelijke meegeeft.

Wie een antwoord geeft en dat even wil testen kan dan gemakkelijker een testfiletje aanmaken om te testen.

Sven Bovin **L^AT_EX en centreren over kolommen in tabel**

Ik heb een tabel waarin bindingsenergieën van twee klassen van chemische bindingen worden vergeleken en zou graag een van de headers over decimaal gecentreerde kolommen laten centreren. Hiertoe gebruik ik de volgende structuur:

```
\documentclass{boek}
\usepackage{booktabs} % om mooiere tabellen te krijgen
\begin{document}

\noindent
\begin{table}
\begin{center}
\begin{tabular}{lr@{${\cdot}$}llr@{${\cdot}$}l}
\toprule % een bovenste lijntje, tgv. booktabs
& \multicolumn{2}{c}{Bindingsenergie} & & \multicolumn{2}{c}{Bindingsenergie} \
\midrule % tussenlijntje, weer booktabs\
C--H & xxx&yy & Si--H & xxx&yy \\
C--O & xxx&yy & Si--O & xxx&yy \\
C--C & xxx&yy & Si--Si & xxx&yy \\
\bottomrule % afsluitend lijntje, ook booktabs
\end{tabular}
\end{center}
\end{table}
\end{document}
```

Maar als ik dat door L^AT_EX haal, worden de headings van de kolommen schijnbaar niet over de decimale kolommen gecentreerd: zij steken t.o.v. de getallen uit naar rechts. Iemand met ideeën hoe ik dit kan oplossen?

Werenfried Spit De multicolumn is wel degelijk gecentreerd. Vervang ‘Bindingsenergie’ maar eens door E_b en je ziet dat dat wel klopt. Waar het misgaat is wanneer de decimaal uitgelijnde kolommen veel smaller zijn dan die gecentreerde kolom; de meest rechtse kolom wordt dan opgerekt. De volgende truc is een beetje rommelig, maar lijnt de decimale punt uit op het midden van ‘Bindingsenergie’:

```
\newlength\rcolw
\settowidth{\rcolw}{Bindingsenergie}
\begin{tabular}{lr@{${\cdot}$}llr@{${\cdot}$}l}
\toprule % een bovenste lijntje, tgv. booktabs
& \multicolumn{2}{c}{ $E_b$ } & & \multicolumn{2}{c}{Bindingsenergie} \\
\midrule %
C--H & xxx&yy & Si--H & \hbox to 0.5\rcolw{\hfill xxx}&yy \\
C--O & xxx&yy & Si--O & xxx&yy \\
C--C & xxx&yy & Si--Si & xxx&yy \\
\bottomrule %=20
\end{tabular}
```

-
- Andrea de Leeuw* **device independent**
- van Weenen* Het was mij al bekend dat hetzelfde dvi-file in dezelfde resolutie afgedrukt op verschillende HP-laserjets een verschillend resultaat opleverde. Op de ene printer is het resultaat veel vetter dan op de andere.
- Nu moest ik echter constateren dat op de HP LaserJet 6MP op de werkplek niet alleen de individuele letters aanzienlijk vetter ogen dan wat de trouwe oude LaserJet II thuis produceert, maar dat ook de breedte van de tekst wel een mm afwijkt. De letters zijn dus echt verschoven. En dat verontrust mij.
- De afdruk op de LaserJet II is met dvihplj op 300dpi gemaakt. Op de 6MP heb ik het met dvihplj op 300dpi, 600dpi en met dvips gedaan. Alle 6MP versies waren qua breedte identiek en duidelijk zichtbaar breder dan de LaserJet II versie.
- Kan iemand me dit uitleggen?
- Piet van Oostrum* De fonts (pk files) moeten voor het juiste printertype gegenereerd zijn. Dus met een metafont mode definitie voor die printer. Sommige printers drukken een pixel veel dikker af dan een andere, en metafont compenseert dan daarvoor. Ook zijn er black-on-white en white-on-black printers en dat maakt nog veel meer verschil uit. (De eerste begint met een leeg “vel” en zet er dan zwarte pixels op, de tweede begint met een zwart “vel” en wist de witte pixels uit. Omdat de pixels i.h.a. cirkeltjes zijn i.p.v. vierkantjes maakt dat nogal wat uit.
- Laserprinters (en inktjetprinters) zijn nogal onnauwkeurige apparaten. Probeer eens een lijn van 10cm af te drukken en meet met een liniaal na. Misschien is je oude printer verkeerd, misschien je nieuwe. Waarschijnlijk heeft het niets te maken met de dvi driver, maar gewoon met de hardware van de printer.
- Edwin Drost* Wie kan mij in dit kader vertellen hoe ik met dvips (meegeleverd bij teTeX onder linux) gebruik kan maken van outline fonts i.p.v. bitmap fonts.
- Piet van Oostrum* dvips -Pcmeps ...
- Als er tenminste een file config.cmeps zit in texmf/dvips/config. Er zit in onze tetex installatie een config.www met iets dergelijks (-Pwww zou je dan moeten gebruiken). Ik heb zelf een config.cmeps met:
- ```
* Additional .map files
*
p +cmfonts.map
p +amsfonts.map
p +lafonts.map
p +cyrfonts.map
p +eufonts.map
p +dcfonts.map
p +dvips-xy.map
p +hoekwater.map
p +xypic.map
*
m 1441536
*
* Default resolution & mode.
*
D 600
M docutech
*
* And send output by default to stdout.
```

\*  
f  
\*

*Martijn Ruppert*

---

### Geheel schreefloze stijl

Is er een stijl die in één keer het gehele document in een schreefloze letter ‘zet’, dus inclusief de sectie-namen, voetnoten, TOC etc... (Oftewel, is er een schreefloos equivalent van Times.sty?).

*Piet van Oostrum*

```
\renewcommand{\familydefault}{cmss}
```

Of als je Helvetica zou willen:

```
\renewcommand{\familydefault}{phv}
```

*Hans Hagen*

```
\setupbodyfont[sansserif]
```

;=)

*Rein Smedinga*

---

### Blokmatrix

Hoe kan in L<sup>A</sup>T<sub>E</sub>X een block-matrix worden gemaakt, waarbij de blokken gescheiden worden door horizontale en verticale gestippelde lijnen (bestaande uit meer dan 3 punten!).

Zijn hier packages voor beschikbaar en zo ja, welke?? Een en ander blijkt in ams(la)tex volgens mijn collega niet te kunnen.

*Piet van Oostrum*

Kijk eens naar het hline package.

David Carlisle heeft een experimenteel package blkarray waar dit soort voorzieningen geïntegreerd is. Te vinden op CTAN in: [tex-archive/macros/latex/contrib/supported/carlisle](http://tex-archive/macros/latex/contrib/supported/carlisle).

*Niels Moes*

---

### Encoding EC-fonts

Ik wil overstappen van de cm fonts naar de ec fonts. Maar de ec fonts zijn 8-bits fonts (256), terwijl de cm er slechts 128 kent. Dan ontstaan problemen met bijvoorbeeld de accenten. Nu kan ik natuurlijk een macro maken die alles omzet, maar ik vermoed dat dit handiger kan. Wie weet welke macro beschikbaar is?

*Hans Renkema*

```
\usepackage[T1]{fontenc}
```

*Sven Bovin*

---

### Labels op A4

Ik kom hierbij terug op een vraag die ik een tijdje geleden al stelde: ik wil labels afdrukken op A4 met behulp van labels.sty. Echter: labels.sty maakt de pagina ‘te groot’, t.t.z.: zet de marges links en rechts op 0. Hoe ik dat moet verhelpen, heb ik intussen gevonden (gewoon wat prutsen met de breedte van de pagina, al kan mijn aanpassing waarschijnlijk wel verbeterd worden, cf. infra)

Nu rest echter nog volgend probleem: labels maakt de pagina te kort, zodat de etiketten te klein worden. Ik heb al geprobeerd van aan de topmargin te knoeien, maar dat helpt niet. En bovendien: zowel de marge bovenaan als die onderaan moeten van 1in naar ongeveer 5mm. Hoe kan ik dat voor mekaar krijgen ?

*bijlage 1: aangepast deel van labels.sty*

```
\textwidth=200mm % origineel: \paperwidth
```

```
\textheight=\paperheight
```

```
\topmargin=-1in
```

```
\headheight=0em
```

```

\headsep=0em
\topskip=0em
\footskip=0em
\oddsidemargin=-20.4mm % origineel: -1in
\evensidemargin=-20.4mm % origineel: -1in
\pagestyle{empty}
\stoptypen}

\blanko\noindent
{\it -- bijlage 2: een voorbeelddocument, 24 labels met Sven --}

\starttypen
% LaTeX2e source file
%
\documentclass[a4paper,12pt]{letter}
\usepackage{mylabels}
\normalsize\sf
\LabelGridtrue
\LabelInfottrue
\TopBorder=24mm
\BottomBorder=20mm
\LeftBorder=10mm
\RightBorder=10mm
%\LabelRows=7
%\LabelCols=2
\begin{document}
%\large\sf

\begin{labels}
Sven
.
.
.
Sven
\end{labels}
\end{document}

```

*Wybo Dekker* Waarom gebruikt je niet label3x8.sty. Heeft PvO eens voor mij gemaakt en die bevat prima. Hier is hij:

```

% label3x8.sty
% usage example:
% \documentclass{letter}
% \usepackage{label3x8}
% \newcommand{\bul}{\hfill\normalsize\bullet\hfill}
% \newcommand{\awa}{% returnaddress
% {\bf\footnotesize NTG \bul{}} Postbus 394
% \bul{} 1740 AJ Schagen\[-.5ex]}
% \protect\rule[2ex]{60mm}{.2mm}\}
% }
%
% \makelabels
% \begin{document}
% \begin{letter}{\awa name1\street1\city1}\end{letter}
% \begin{letter}{\awa name2\street2\city2}\end{letter}

```

```

% \begin{letter}{\awa name3\street3\city3}\end{letter}
% \begin{letter}{\awa name4\street4\city4}\end{letter}
% \begin{letter}{\awa name5\street5\city5}\end{letter}
% \end{document}

\renewcommand*{\startlabels}{\labelcount\z@
\pagestyle{empty}%
\let\@texttop\relax
\topmargin -1in
\headsep \z@
\headheight \z@
\oddsidemargin -20.4mm
\evensidemargin -20.4mm
\textheight 297mm
\@colht\textheight \@colroom\textheight \vsize\textheight
\textwidth 20cm \hsize\textwidth
\ifcase \@ptsize\relax
\ normalsize
\or
\ small
\or
\ footnotesize
\fi
\baselineskip \z@
\lineskip \z@
\boxmaxdepth \z@
\parindent \z@
\relax}

\renewcommand*{\mlabel}[2]{%
\parbox[b][37mm][c]{60mm}{\strut\ignorespaces #2}%
\hskip0pt\penalty-100\hskip10mplus 5mm}

```

*Maarten Gelderman*

### **Lexmark**

Wellicht een wat oneigenlijke vraag voor dit forum, maar mijn printer staat op het punt te overlijden en ik ben op zoek naar een nieuwe. Heeft iemand ervaring met de Lexmark E310 (nummer kan verkeerd zijn) dat is een printer van iets meer dan hfl 800 die postscript schijnt te eten. En zo ja: eet het ding echt postscript en hoeveel extra geheugen moet erin (of is het überhaupt een soort winddoos).

*Taco Hoekwater*

Deze ken ik niet (nou ja, net de PPD zitten lezen), maar een paar andere Lexmark Optra's wel. In het algemeen geldt voor de Optra's:

- ▣ PS is een (flink!) stuk trager dan de PCL mode, maar in het algemeen erg betrouwbaar (de Optra's gebruiken allemaal dezelfde interpreter, en er zitten ook een paar high-end modellen in). De PS mode is beslist beter dan die van HP.
- ▣ 2Mb ram is veel veel te weinig. +4Mb is echt nodig om te zorgen dat grote plaatjes het ook doen in de hogere resoluties, maar +16Mb is echt veilig en raad ik aan (wegens geheugen gebruik van PS fonts).
- ▣ Zorg ervoor dat je de input buffer in de printer zo klein mogelijk maakt. (dit is algemeen voor al dit soort printers: een te grote buffer betekent soms dat je de grootste moeite hebt jobs te cancellen)
- ▣ De Ethernet interface kaartjes zijn iets van F800, en beslist de moeite waard.

Specifiek voor deze E310:

- Je krijgt 39 PS fonts, maar het zijn wat oudere versies die niet zo perfect zijn als de laatste nieuwe van Adobe.
- dat geschermd met 1200dpi betekent dat ie figuren ook in 851pi kan printen, de fonts blijven altijd 600x600 (geen enhancement of smoothing)
- In PS mode kun je rekenen op ongeveer 4-5 ppm kale tekst, 1-2 met grote figuren (met een opstarttijd van pakweg een minuut, \*niet\* 19sec)
- De flash-memory kaartjes zijn bruikbaar voor permanente download van fonts.
- Volgens mij zijn de cartridges Lexmark-specifiek.

Het lijkt me een verantwoorde aanschaf, maar dat hangt er wel een beetje van af hoe duur die RAMmetjes zijn. Als ze standaard RAMS gebruiken ben je goedkoop klaar (maar dat zal dus wel niet).

*Hans Hagen* Aangezien GS de meeste printers wel kan aansturen, zou ik (voor thuisgebruik) mijn oog laten vallen op een Epson Photostyler (700\*1400 of hoger) of een Epson A3 printer. Vooral de Photostyler is een opvallend goeie kleurenprinter (koop ik nog eens) en bovendien tegenwoordig betaalbaar. Ik denk dat de goedkope PS printers altijd een beetje zullen tegenvallen (klonen, te weinig geheugen, traag en zo, maar ik kan me vergissen).

---

*Sven Bovin* **LaTeX: veranderen v/d nummer van een figure**

Hoe kan ik in een document de nummering van figuren laten beginnen bij een willekeurig getal? Ik zou namelijk om technische redenen mijn figuurnummering willen laten starten op 2. Is dat een `\renewcommand` of een `\setcounter`, en welke juist?

*Paul Huygen* `\addtocounter{figure}{1}`

---

*Edwin Roos* **On the fly previewen**

Does anybody know if there is software in which it is possible to enter text in one window and look at the output produced by LaTeX in another window. That output needs to be generated on the fly.

*Wybo Dekker* You can do that with `prv` (which is on the TeXLive CD under `support/prv`) by running

```
$ prv -pvc yourfile[.tex] &
$ vi yourfile.tex # or use another editor
```

The first command runs LaTeX on `yourfile.tex` as many times as necessary and previews the dvi-file with `xdvi` in the background. Every time you write your vi-buffer to disk, your `xdvi` window will get updated. If you like it, ask my latest version.

*Sven Bovin* If you're using Linux (possibly other Unix variants as well), you could use LyX/KLyX. There is, of course, a catch: rumour has it that the output of LyX is not strictly LaTeX compliant, but I cannot tell you whether that is true or not, because I've never used it myself.

*Martin Schroeder* TeXtures and Y&Y TeX claim to be able to do this.



*Lex Sijtsma***hyperref**

Ik gebruik het package hyperref als volgt:

```
KB-CAT can be seen in action at
\href{http://www.konbib.nl/zoek/kbc\sqry_el.html}%
{www.konbib.nl/zoek/kbc\sqry_el.html}.
At the moment it contains primarily information about
electronic publications.
```

Dit levert 4 paar nagenoeg identieke foutmeldingen op:

```
! Undefined control sequence.
<argument> ...ct \global \let \OT1\textunderscore
\unhbox \voidb@x \kern
.06...
1.341 ...1}{www.konbib.nl/zoek/kbc\sqry_el.html}
. At
?
! Undefined control sequence.
<argument> ...hrule width.3em}\OT1\textunderscore
sqry\protect \global \let
...
1.341 ...1}{www.konbib.nl/zoek/kbc\sqry_el.html}
```

Enig idee wat ik fout doe?

P.S. Zonder \ heb ik al geprobeerd. Werkt ook niet.

*Piet van Oostrum*

Kennelijk kan hyperref niet met \\_ overweg. Je zou dit als een bug kunnen melden.

Een oplossing kan zijn om de \\_ in de URL te vervangen door %5F. Of zoiets:

```
{\catcode_ =12\href{http://www.konbib.nl/zoek/kbc_sqry_el.html}%
{www.konbib.nl/zoek/kbc\sqry_el.html}}
```

Mocht je dit in een macro willen doen dan moet je even goed opletten dat de catcode op het juiste moment wordt veranderd.

```
\newcommand{\myhref}{\begingroup\catcode_ =12\myhrefii}
\newcommand{\myhrefii}[2]{\href{#1}{#2}\endgroup}
\myhref{http://www.konbib.nl/zoek/kbc_sqry_el.html}%
{www.konbib.nl/zoek/kbc\sqry_el.html}
```

*Willi Egger***Context: aanmaken van een A4 met een 2\*4 lay-out**

Ik poogde vanmiddag een document van twee a4tjes zo op te maken, dat ik het als boekje kon vouwen. hierbij gebruikte ik

```
\setuppapersize[A6]A4]
\setuparranging[2*4, doublesided]
```

Het resultaat was wel een vel met paginas in een inslagschema. Echter is het niet mogelijk het schema zo te vouwen, dat de volgorde klopt. Dit komt omdat de pagina's 3 en 6 niet op de goede plaats zitten (van kant verwisseld, 6 hoort links boven en 3 rechts boven te staan). Hoe kan ik dit oplossen? – Waar ligt het probleem?

In verband met mijn pogingen een aangepaste hoofdregel in mijn dokument te bouwen, struikelde ik tevens over het volgende: Ik wil graag een haarlijn onder de hoofdregel laten zetten. Hiervoor gebruik ik de volgende regels aan het begin van het dokument:

```
\setupbodyfont[ber,phv,ss,8pt] %Helvetica
\setupheader[style=slanted,after=\nowhitespace\hairline]
\setuppagenumbering[alternative=doublesided, location={header,margin}]
```

Na compilatie verschijnt de haarlijn echter vanaf de papier-rand tot 'ergens' onder de hoofdregel-tekst. Hoe krijg ik die haarlijn wel op de goede plaats met een lengte van de tekstbreedte?

*Hans Hagen* Voor wat betreft de haarlijn:

```
\setupbackgrounds
 [header]
 [text]
 [bottomframe=on,
 frame=off]

\setuplayout
 [header=20pt,
 headerdistance=1cm]
```

En over de 2\*4-layout: Ik weet niet precies hoe een drukker dit doet, maar hoor graag wat de officiële versie is. Zonodig, voeg ik een alternatief toe. Je kunt eens proberen de onderstaande macro toe te voegen en 3 en 6 te verwisselen,

```
\def\pusharrangedpageEIGHT#1%
 {\doglobal\increment\arrangedpageN
 \reportarrangedpage\arrangedpageN
 \ifcase\arrangedpageN
 \or \handlearrangedpageXY{#1}{180}{0}{1}\arrangedpageA % 1
 \or \handlearrangedpageXY{#1}{180}{1}{1}\arrangedpageB % 2
 \or \handlearrangedpageXY{#1}{0}{0}{0}\arrangedpageB % 3
 \or \handlearrangedpageXY{#1}{0}{0}{0}\arrangedpageA % 4
 \or \handlearrangedpageXY{#1}{0}{1}{0}\arrangedpageA % 5
 \or \handlearrangedpageXY{#1}{0}{1}{0}\arrangedpageB % 6
 \or \handlearrangedpageXY{#1}{180}{0}{1}\arrangedpageB % 7
 \or \handlearrangedpageXY{#1}{180}{1}{1}\arrangedpageA % 8
 \poparrangedpages
 \fi}
```

Je ziet, het ziet er niet eens zo ingewikkeld uit, Overigens, in de laatste texexec kun je eerst runnen

```
texexec --noarrange filename
```

en een laatste run

```
texexec filename
```

Dat scheelt wat wegprocenten.

*Marcel Heldoorn***Footnotemark en footnotetext**

Ik heb een probleem met footnotes:

```
% minimal test file
\documentclass{article}
\begin{document}
\begin{table}[btp]
\begin{minipage}[]{\textwidth}
\centering
\begin{tabular}{|cccc|}
\textbf{first} & \textbf{second} & \textbf{third} & \textbf{fourth}
\\ \hline
Dit\footnotemark[1] & is\footnotemark[2] & een\footnotemark[3] &
test\footnotemark[4] \\
Dit\footnotemark[1] & is\footnotemark[2] & een\footnotemark[3] &
test\footnotemark[4] \\
\end{tabular}
\footnotetext[1]{footnote I}
\footnotetext[2]{footnote II}
\footnotetext[3]{footnote III}
\footnotetext[4]{footnote IV}
\end{minipage}
\centering
\caption{test}\label{test}
\end{table}
\end{document}
% \end minimal test file
```

Ik wil graag de footnotes met een `\alph` label, maar in de tabel worden cijfers gebruikt.*Piet van Oostrum*Zet even `\renewcommand{\thefootnote}{\alph{footnote}}` achter `\begin{minipage}`*Marcel Heldoorn*

Wat is dit toch een geweldige mailing list. Binnen een uur antwoord en de juiste oplossing!

*Stefan Goeman***Welke printer?**

Voor thuis gebruik zou ik mij een printer willen aanschaffen. Ik maak regelmatig documenten in  $\text{\LaTeX}$  en ik ben nu sinds kort ook aan mijn eindverhandeling begonnen.

Ik zou graag het eindresultaat (postscript file gemaakt met dvips) kunnen afprinten bij mij thuis.

Gezien ik absoluut geen verstand heb van printers, zou ik wel wat raadgevingen kunnen gebruiken. Ik begrijp zeer goed dat prijs en kwaliteit in deze materie nauw samenhangen. Mijn budget is echter ook beperkt natuurlijk. Het budget dat ik maximaal wens te besteden is 1500 fl. (of 30.000 - 35.000 bfr.).

*Roland Smith*

Ik denk dat printers met een ingebouwde postscript interpreter niet in deze prijsklasse vallen. :-)

Je zult daarom een printer moeten zoeken waar Ghostscript mee kan werken. Een goede plaats om te starten zou kunnen zijn: [http://gatekeeper.picante.com/~gtaylor/pht/printer\\_list.cgi](http://gatekeeper.picante.com/~gtaylor/pht/printer_list.cgi) Deze pagina gaat over printers die onder Linux te gebruiken zijn. Maar aangezien veel Linux gebruikers Ghostscript als 'printer-driver' gebruiken, is de lijst ook wel bruikbaar als je Ghostscript op een ander OS gebruikt.

Meer informatie over Ghostscript kun je vinden op <http://www.cs.wisc.edu/~ghost> of <http://www.ghostscript.com>. Zelf heb ik een Deskjet 500 gehad, en heb nu een Laserjet 5L. Beide zijn goede zwart/wit printers. De deskjet had als bezwaar dat de patronen vrij duur zijn, en dat ze snel uitdroogden. Over de Laserjet ben ik heel tevreden.

Koop liever geen printer die aangemerkt staat als 'winprinter'. Als je geen windows gebruikt koop je hiermee een kat in de zak, en ook Ghostscript zal er geen driver voor hebben.

*Jozef Pijnenburg*

---

### Afbreken in section headers

ik wil een lange section header op een bepaalde plaats afbreken, zonder dat die afbreking doorwerkt in de toc en/of pageheadings. Ik heb al wat geprobeerd met `\linebreak`. Dat levert echter niet het gewenste resultaat. Is dat ooit iemand wel gelukt ?

*Peter van der Wijden*

```
\section{Hier wil ik een afbreking plaatsen}%
{Hier wil ik een afbreking plaatsen,\\%
maar ik wil niet dat dit in de table of contents komt.}
```

*Lico Hoekema*

---

### Roterende plaatjes

Met behulp van de suggestie van PvO om mbv van het fancyhdr package een plaatje in te voegen en dat op elke volgende pagina een stukje geroteerd te krijgen heb ik de volgende oplossing bedacht:

Ik heb 8 plaatjes gemaakt die telkens 45 graden t.o.v elkaar gedraaid zijn. Ik wil alleen op de oneven pagina's een plaatje hebben. Als ik bij plaatje 9 dreig te komen, moet de teller weer teruggezet worden naar 1.

```
\newcounter{roteer}
\setcounter{roteer}{1}
\renewcommand{\chaptermark}[1]{\markboth{\shorttitle}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\lhead[\fancyplain{}]{\bfseries\thepage}%
{\fancyplain{}{\bfseries\rightmark}}
\rhead[\fancyplain{}]{\bfseries\leftmark}%
{\fancyplain{}{\bfseries\thepage\raisebox{0cm}[0pt][0pt]%
{\parbox{0cm}{\vspace{4cm}}{\value{roteer}}}%
\addtocounter{roteer}{1}}%
\ifthenelse {\value{roteer} = 9} {\setcounter{roteer}{1}}{}%
}}}\cfoot{}
```

Het probleem zit hem in `\value{roteer}`. Ook als ik hier `\value{page}` van maak krijg ik een foutmelding (Missing number, treated as zero). Als ik hier gewoon tekst zet gaat het goed. Uiteindelijk met `\value{roteer}` vervangen worden door

```
\includegraphics*[width=2cm]{roteer\value{roteer}.eps}
```

*Piet van Oostrum*

`\value{roteer}` is een getal, en dat mag je alleen gebruiken op een plaats waar ermee gerekend wordt. Maar je wilt het hier als tekst gebruiken en dat kan niet. je moet dan de tekstversie gebruiken en die is `\theroteer`. Of `\thepage` als het over het paginanummer gaat. Dat geldt ook voor de `\includegraphics`. De fancyhdr documentatie heeft daar ook `\thepage` staan.

*Arne Scollaert*

---

### Marges

Wat moet ik doen in L<sup>A</sup>T<sub>E</sub>X om op een relatief eenvoudige manier mijn marges (van een A4-blad) naar eigen believen (tz naar de wensen van mijn baas) aan te passen.

*Geert Stremersch* In L<sup>A</sup>T<sub>E</sub>X in de preamble bijvoorbeeld:

```
\textheight 23.8cm
\textwidth 17.2cm
\topmargin 0mm
\oddsidemargin -5mm
\headsep 0cm
\hoffset 0cm
\voffset 0cm
\headheight 0cm
```

*Paul Huygen* Gebruik de lengtes `\oddsidemargin`, `\evensidemargin` en `\textwidth`. Bij het bepalen van de marges moet je in het oog houden dat het nulpunt op 1in van de linkerkant van het papier ligt. Als de rechtermarge bijvoorbeeld 2 cm moet zijn, zet dan in de preamble `\setlength{\oddsidemargin}{5mm}` Zie het L<sup>A</sup>T<sub>E</sub>X boek van Lamport, of The L<sup>A</sup>T<sub>E</sub>X Companion van Goossen et al, bladzijde 85.

---

*Lico Hoekema* **Headers en footers**

Weer een probleempje met headers in L<sup>A</sup>T<sub>E</sub>X. Ik wil op mijn oneven pagina een header hebben die hoofdstuknummer/sectionnummer met daarachter de sectienaam. Bovendien moet er nog een lijn onder komen die precies even lang is als de bovenstaande tekst. Op de even pagina's moet de lengte v/d lijn gewoon `\headwidth` zijn. Mijn probeersel werkt (weer eens) niet:

```
\settowidth{\mylen}{\thesection \sectionname}
\renewcommand{\headrule}{\vbox to 0pt{\hbox to \mylen{\hfill}\vss}}
\sectionname bevat niet de naam v/d sectie?? Bovendien heb ik nu nog niet de lengte vd
lijn afwisselend gekregen.
```

*Piet van Oostrum* Erg fraai kan ik het niet noemen. Maar ik zal maar toegeven dat ik daar niet over ga.

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhead[LO]{\underline{\strut\thesection\rightmark}}
\fancyhead[RO]{}
\renewcommand{\headrulewidth}{\ifodd\value{page}0pt\else 0.4pt\fi}
```

---

*Maarten Wisse* **Halftone rasters met dvips**

Nu ik met ConTeXt zo ongeveer dezelfde DTP gekke dingen kan doen als met Corel Ventura, geniet ik van DTP met Linux. Maar, omdat ik tot op heden nog op papier aanlever, moet ik plaatjes hebben in een soort krant-half-tone raster. Anders treedt bij het drukproces enorm veel contrastverlies op. Ik weet het, het is niet de manier om te drukken, maar het lijkt op dit moment het goedkoopst.

In Ventura kan ik bij het printen een half-tone raster opgeven van zoveel lpi, waardoor alle plaatjes van zo'n raster worden voorzien. Mijn vraag is of dat middels dvips ook kan, en zo niet, hoe ik het dan wel moet doen. Mijn plaatjes komen uit GIMP, worden opgeslagen in EPS en vervolgens opgenomen in ConTeXt.

- Siep Kroonenberg* Je kunt aan dvips een header file opgeven waarin je een raster definieert:
- ```
dvips -h raster.pro filenaam
```
- waar raster.pro bijvoorbeeld als inhoud heeft:
- ```
%!
/bop-hook {53 45{dup mul exch dup mul add 1. exch sub}setscreen} def
```
- bop-hook is een functie (hook) die aan het begin van elke pagina wordt aangeroepen. 45 is de rasterhoek, 53 de rasterfrequentie. Speel wat met deze laatste parameter. bop-hook krijgt het paginanummer als parameter mee en wil die ook weer terughebben, maar daar heb je in dit geval niets mee te maken.
- 
- Arne Schollaert* **import uit excel**
- Ik wil een grafiekje, in excel samengesteld op basis van gewone data, importeren in L<sup>A</sup>T<sub>E</sub>X. Hoe moet ik daarvoor mijn figuur in excel saven?
- Siep Kroonenberg* Zucht. Gooi excel het raam uit. Of als je echt denkt iets bruikbaar uit M-sof te kunnen persen: gebruik de eps export optie van een PostScript printer driver. Een andere tip die ik op deze lijst voorbij zag komen was om mbv. excel's macrotaal gnuplot invoer te genereren.
- Arne Schollaert* Ik heb mijn excel-grafiek in een ander programma geplakt, in mijn geval CorelDraw, en dit dan als eps geëxporteerd naar L<sup>A</sup>T<sub>E</sub>X. Maar bij het previewen in windvi loopt het fout. Het compileren lukt, maar bij de preview krijg ik volgende foutmelding:
- ```
Error: /undefined in A?OE Operand stack:
```
- etc. Weet iemand daar raad mee, of kan iemand mij vertellen wat ik als alternatieven heb?
- Paul Huygen* Windows programma's genereren (bijna) allemaal rare postscript. Vaak kun je het repareren door alle rare rommel die aan het begin en eind staan er uit te gooien (d.w.z. voor %!ps . . . en na %%Trailer. Zelf begin ik er niet meer aan. Ik laat Windows programma's plaatjes in portable pixelmap formaat, of anders in gif, tiff of bmp formaat genereren, en converteer die dan (onder Unix) met scripts die ik daarvoor geschreven heb.
- Maarten Wisse* Dit ziet er typisch uit als een onder Windows gemaakte eps die weer eens niet normaal geproduceerd is. Waarschijnlijk is de uitvoer in CorelDraw niet helemaal zoals het hoort. Dat is gebruikelijk bij windows-programma's, als het over postscript gaat. Ik weet niet of je Staroffice bij de hand hebt, maar daarin ben ik in staat om een excel-grafiek te kopiëren naar een bitmapgeoriënteerd programmaatje, en kan ik die weer opslaan als een jpg/png of zoiets. Als je daar bent, kun je met Imagemagick die file converteren naar eps en nog eens proberen. Ik ken Imagemagick niet zelf, maar het staat op mijn T_EX Live 4 als beschikbaar onder dos-systemen en uit de help blijkt dat het dit aan zou moeten kunnen via convert.exe. Misschien kunnen anderen vertellen hoe dat moet.
- De moraal van het verhaal is dus: zorg dat je chart een bitmapfile wordt, want dan kan Imagemagick er eps van maken. Groeten,
- Siep Kroonenberg* Als je niet de allernieuwste dvips hebt dan moet je zorgen de eps ZONDER preview-header op te slaan.
- Ik denk dat je op deze manier een lage-resolutie bitmap krijgt. Als je wel een vectorplaatje krijgt moet je erop letten dat ofwel alle letters worden 'getekend' ofwel alleen Times/Helvetica/Symbol wordt gebruikt. Kijk dus even de eps exportopties van CorelDRAW door.
- Een eps file via een PostScript printerdriver is een veiliger methode.

Werenfried Spit Windows programma's genereren inderdaad (bijna) allemaal rare postscript, maar wanneer je de printerdriver van Adobe gebruikt in plaats van die van MS dan gaat het toch heel aardig. Je kunt die driver bij Adobe downloaden en dan gewoon ipv je MSdriver gebruiken.

Gert Bartholomeeusen **EPS Watermerk**

Hoe breng ik een watermerk aan in een L^AT_EX-tekst? De figuur die ik als watermerk wil is een Encapsulated Postscript file.

Wat is de meest eenvoudige manier om dit te doen? Ik heb er de L^AT_EX-companion eens op nageslagen maar ben daar niet veel wijzer van geworden.

Hendri Hondorp Een kant en klaar antwoord heb ik dus niet, maar wel een aanwijzing.

Op blz 429-430 van de L^AT_EX Graphics companion wordt een uitleg gegeven over het toevoegen van bijvoorbeeld het woord DRAFT diagonaal over de bladzijden heen. Dit is in PostScript, dus je hebt dvips nodig. Dvips heeft een bop-hook 'macro' die bij elke nieuwe pagina aangeropen wordt.

In dit boek staat:

Maak file secret.pro:

```
userdict begin
/bop-hook{gsave
200 150 translate
65 rotate
/Times-Roman findfont 144 scalefont setfont
0 0 moveto
0.7 setgray
(SECRET!) show
grestore} def end
```

Het commando

```
dvips -h secret.pro test.dvi -o test.ps
```

geeft een PostScript file test.ps met als kenmerk een tekst SECRET van links onder tot rechts boven (in grijs). De inhoud van jouw PostScript file zou ook in zo'n /bop-hook moeten komen.

Ook staat er dat er een mogelijkheid is om een config file te maken en deze te gebruiken: config.watermerk:

```
E echo /bop-hook userdict begin \{ gsave > xxx.txt
E cat watermerk.eps >> xxx.txt
E echo show grestore >> xxx.txt
E \} def end >> xxx.txt
h xxx.txt

dvips -Pwatermerk test.dvi -o test.ps
```

Misschien moeite met het vinden van files (paden) maar zo iets zou ook kunnen denk ik. Ik ben er niet zeker van, dus je moet maar eens iets proberen.

Arne Schollaert **Specifieke plaats**

Dit is vermoedelijk een zeer primaire en domme vraag, maar het lukt me toch niet. Ik tracht een tabel, gewoon in L^AT_EX opgemaakt, op een bepaalde plaats in m'n tekst te zetten, maar latex zelf zet het telkens elders. Met andere woorden, hoe kan ik L^AT_EX verplichten een tabel op een welbepaalde plaats te laten staan?

Sven Bovin De tabel niet in binnen een `\begin{table} ... \end{table}` constructie plaatsen zou kunnen helpen (dan is het geen float meer). Beter is waarschijnlijk het package `float` te gebruiken. Dit package laat toe de placement specifier `H` te gebruiken, die wil zeggen: hier en nergens anders. Voordeel is dat de tabel automatisch genummerd word, een caption kan krijgen, ... terwijl je toch een hier-en-nergens-anders- bevel hebt.

Liesbeth van der Plas

Tekening in document

Het plaatsen van een tekening in een Context document lukt mij niet. Ik heb de tekening: `poes.eps` opgeslagen in `C:\4tex5.0\figuren`. Het instelgebied plus de rest van mijn MED-file heeft de volgende inhoud:

```
\stelwitruimtein[groot]
\stexternfigurenin[gebied=C:/4tex5.0/figuren]
\gebruikexternfiguur[plaatje][poes]
[type=eps, breedte=.4\tekstbreedte]
\starttekst
...
\plaatsfiguur
[hier,forceer]
[fig:poes]
{Een poes.}
{\naam{poes}}
...
\stoptekst
```

De figuurrand plus het figuuronderschrift wordt wel geplaatst, maar binnen de omlijsting staat: "niet gedefinieerd". Ook `[gebied=/figuren]` werkt niet.

Hans Hagen

```
\gebruikexternfiguur
[plaatje][poes][type=eps,breedte=.4\tekstbreedte]
```

Is okee, maar het `type=eps` is niet nodig. ConTeXt zoekt zelf uit wat het beste formaat is (wat in geval van pdf uitvoer geen eps maar pdf is). Je figuur heet niet `poes` maar `plaatje`. Je moet dus `\naam{plaatje}` opgeven, of beter: `\externfiguur[plaatje]`, dus:

```
\plaatsfiguur
[hier,forceer][fig:poes]
{Een plaatje van een poes.}
{\externfiguur[plaatje]}
```

Het idee achter de logische naam `plaatje` is dat je afmetingen consistent kunt definiëren. Als je zegt:

```
\gebruikexternfiguur[nog een plaatje][hond][plaatje]
```

dan krijg je met:

```
\externfiguur[nog een plaatje]
```

een hond te zien die net zo is geschaald als poes. Dus, als je in plaats van instellingen een naam van een eerder gedefinieerd figuur opgeeft, dan erven de eigenschappen over. Dit kan handig zijn als je bv 20 plaatjes hebt, verspreid in je document, en je wilt ze gelijkwaardig behandelen.

-
- Maarten Wisse* **Van postscript naar druk**
 Zoals wij allen ben ik natuurlijk rotsvast overtuigd van het feit dat \TeX het mooiste zetwerk oplevert. Alleen levert die overtuiging nog wel eens wat problemen op met drukkers, zoals ik vanochtend weer gemerkt heb.
 Mijn probleem was als volgt, of probleem, volgens mij was dat er niet eens, maar volgens de drukker wel. Ik heb in ConTeXt een boekje opgemaakt (A5) en belde mijn drukker met de vraag hoe hij dat precies wilde hebben. Ik dacht dan met name aan de vlakverdeling en arrangering in ConTeXt, die je daar zo prachtig in kan stellen. Antwoord: je kunt helemaal niet elektronisch aanleveren in postscript of pdf, want wij werken alleen met QuarkXpress. Lever maar gewoon in Word aan, of als het dan per se moet, camera-ready op papier. Word leek met niet zo'n goede gedachte, dus wordt het nu waarschijnlijk papier.
 Vraag: hoe doen jullie dat, of beter gezegd, hoe zou mijn drukker het kunnen doen? Zijn argument is dat het uitdraaistation waar hij zijn films laat maken alleen maar Xpress-output slikt. Is daar iets aan te doen?
- Piet van Oostrum* Andere drukker kiezen.
- Maarten Wisse* Dit antwoord is natuurlijk in principe afdoende, maar dat betekent dus dat er ook uitdraaistations zijn die wel postscript of pdf slikken? Hoe zou ik erachter kunnen komen waar die zich bevinden? Dan kan ik mijn drukker daarmee in contact brengen wellicht. Groetjes,
- Johan Vromans* Grote kans dat hun Xpress keurig PostScript produceert voor de filmmachine. Of anders heeft men een speciale 'printer' driver voor deze machine, maar die kan dan waarschijnlijk ook worden gebruikt door Acrobat of GhostView. Het is het uit handen geven van het productieproces waar men zenuwachtig van wordt.
- Maarten Wisse* Dat laatste klopt in ieder geval ja, want de opmerking viel keer op keer dat ze dan niet meer wisten hoe het precies moest en dat ze er dus geen tijd voor hadden.
- Piet van Oostrum* Zoals je zelf hebt aangegeven heeft het meer met de onwil of onkunde van de drukker te maken. Als je niet aan deze drukker vast zit, zou ik een andere zoeken. Ik weet er zelf geen, want ik heb niks te drukken wat uitgaat boven een copieermachine, maar de redactie van MAPS weet misschien wel een adres.
- Siep Kroonenberg* Als een drukker benauwd is van PostScript- en pdf bestanden dan ga ik hem niet mijn PostScript- en pdf bestanden toevertrouwen.
- Taco Hoekwater* Ik weet wel een paar drukkers, maar ik ben niet op kantoor dus ik heb die adressen niet bij de hand, sorry. Het is niet zo heel ongewoon dat drukkers in ieder geval met willekeurige PostScript files overweg kunnen hoor. Elke derde is raak denk ik.
 Die drukker bedoelde te zeggen dat hij alleen weet hoe hij Quark files naar de RIP moet sturen (en dus niet dat de rip het niet kan). Dat is feitelijk wat anders, maar uit ervaring weet ik dat je dat beter ook niet kun proberen uit te leggen.
 Gewoon wat drukkers bellen en vragen of ze overweg kunnen met kale PS files. Als ze dan ja zeggen, kun je daarna vragen of PDF ook kan. Niet eerder, wat Quark heeft een import filter voor PDF, en dat ding is zo vrakkig dat je jezelf dan nog veel meer problemen op de hals haalt.

- Maarten Wisse* Bedankt voor de informatie. Het maffe van het hele probleem is nog wel dat zo'n drukker je eerst voor doet komen dat je niet wijs bent om met een postscript file bij hen te komen. Dat zou iets van de toekomst zijn. In feite is het dus hun eigen starheid, waarbij deze drukker ook nog aangaf de monopolie van Quark te betreuren, maar daar werken ze op deze manier natuurlijk zelf wel aan mee. Althans voorzover men niet gewoon naar een andere drukker gaat, zoals ik in het vervolg zal doen.
- Piet van Oostrum* Ik zou je willen aanraden om die drukker dan ook te laten weten dat dit de reden is dat je een ander zoekt. Misschien brengt hen dat op andere gedachten.
- Wybo Dekker* Ik heb goede ervaringen met Datawyse, Maastricht (info@datawyse.nl). Aanleveren als pdf-file.
- Andrea de Leeuw van Weenen* Ik heb mijn boek in PS aangeleverd bij Ridderprint, Pottenbakkerstraat 15-17, 2984 Ridderkerk (Tel 0180-463962). Afgezien van het feit dat ze de tekst op een bepaalde positie van de pagina wilden hebben geen problemen. Ze hebben ook het 4TeX for Windows boek gedrukt.

Bessem, J.M. **Uitlijnen met enumerate**

Graag zou ik een oplossing hebben voor het volgende 'uitlijn'-probleem:

```
\begin{enumerate}
\item <lange tekst0>\
\Box$ <tekst1> \hfill $\boxtimes$ <tekst2> \hfill $\Box$ <tekst3>
\item <lange tekst 4>\
\boxtimes$ <tekst5> \hfill $\Box$ <tekst6> \hfill $\Box$ <tekst7>
\item
enz.
```

waarbij de vierkantjes van de ene regel steeds netjes boven de vierkantjes van de volgende regel komen te staan. Zoals ik het nu heb werkt dat niet goed.

- Piet van Oostrum* Misschien is dit iets om mee te beginnen:

```
\newcounter{entrynr}
\newcommand{\entry}[1]{\multicolumn{7}{1}{\refstepcounter{entrynr}
\makebox[1em]{\theentrynr.} #1}}

\begin{tabular}{p{1em}clclcl}
\entry {<lange tekst0>}\
& $\Box$ & <tekst1> & & $\boxtimes$ & <tekst2> & & $\Box$ & <tekst3>\
\entry {<lange tekst 4>}\
& $\boxtimes$ & <tekst5> & & $\Box$ & <tekst6> & & $\Box$ & <tekst7>\
\end{tabular}
```

Geert **Block comments**

Bestaat er in latex2e een snellere manier om grotere stukken tekst uit te commenten dan voor elke lijn een %-teken te plaatsen ?

- Piet van Oostrum* Met een fatsoenlijke editor is een % voor elke regel zeer snel te doen. Als er geen ongeneste \if constructies inzitten kun je er \iffalse voorzetten en \fi erachter.
Anders kun je het comment of xcomment of verbatim package gebruiken, met \begin{comment}...\end{comment}

- Hans Hagen* `\long\def\StartForgetAboutIt#1\StopForgetAboutIt{}`
`\starttypen`
`\StartForgetAboutIt`
 weet ik wat voor'n tekst
`\StopForgetAboutIt`
In context, zonder geheugenprobleem:
`\definieerblok[bagger] \verbergblokken[bagger]`
`\beginvanbagger`
 weet ik wat voor'n tekst
`\eindvanbagger`
-
- Stefan Goeman* **Bitmaps en eps**
 Als ik van een bitmap foto een eps-file maak (via Coreldraw), dan merk ik dat de eps-file veel groter is (minstens x2) dan de bitmap file. Is er een mogelijkheid om dit te verhelpen?
- Piet van Oostrum* Ik neem aan dat je met groter bedoelt het aantal bytes, niet de afmetingen van het plaatje. Je kunt de omzetting beter doen met ImageMagick. Die heeft een eps2 mode die gecomprimeerde eps maakt. Dit kan overigens alleen gebruikt worden met Postscript versie 2 of hoger, maar alle gangbare moderne printers hebben dat. De eps file is dan maar iets groter dan de bitmap (of als de bitmap niet gecomprimeerd is, misschien zelfs wel kleiner). Imagemagick vind je op <http://www.wizards.dupont.com/cristy/ImageMagick.html>
- Stefan Goeman* Interessant misschien voor mensen die eps-files naar pdf-files willen omzetten, is het programmaatje `epstopdf.exe`; zie <http://tug.org/applications/pdftex/epstopdf>
-
- Wybo Dekker* **Fonts / design sizes**
 Een vriend van mij is in L^AT_EX bezig een menukaart te maken en is gecharmeerd van fonts zoals `cmfi` en `cmdunh`. Zo'n font kun je gebruiken na bv `\newfont{\font}{cmdunh10}` maar als je daarna `\large` of `\small` gebruikt krijg je een heel ander font (welk eigenlijk?). Komt dat omdat er maar 1 design size van zo'n font bestaat, en maakt dat zo'n font niet erg beperkt bruikbaar?
- Hans Hagen* Gewoon direct doen:
`\font\LargeFunnyFont=cmdunh10 at 30pt`
`\LargeFunnyFont`
 Hier eet u smakelijk tijdens
 het roken van een DunHill.
-
- André de Boer* **De man als symbool**
 Kan iemand mij vertellen hoe ik in T_EX de symbolen voor 'man' en 'vrouw' kan invoeren. (Rondje met pijltje en rondje met een kruis.)
- Piet van Oostrum* `\usepackage{wasysym}`
`\male \female`

- André de Boer* Bedankt voor je snelle reactie Piet. Maar ik stuit op het volgende probleem:
- ```
cannot find wasy source file.
hbf2gf -q-g wasy10 300
MAKETFM: don't know how to make wasy10
```
- Piet van Oostrum* Vreemd, dat ie hbf2gf gebruikt. Heb je een directory texmf/fonts/source/public/wasy met .mf files erin?
- André de Boer* Nee wel een D:\texmf\fonts\TFM\public maar daar zit niets in.
- Piet van Oostrum* Dan moet je ze even ophalen, bijv van ftp://ftp.ntg.nl/pub/tex-archive/fonts/wasy2.zip of van een CDROM die je bij de MAPS hebt gekregen.
- André de Boer* Vergeef mijn onkunde. Ik heb de fonts ergens gezet, MikTeX opnieuw geïnitieerd maar nog steeds:
- ```
! Font U/way/m/n/10=wasy10 at 10.0pt not loadable:
Metric <TFM> file not found
```
- Peter van der Wijden* Heb je de MikTeX structuur aangehouden? Dan zou het als volgt kunnen werken, en bij mij werkt het: plaats
- ```
de tfm-files in: localtexmf\fonts\TFM\public\wasy
de mf-files in: localtexmf\fonts\source\public\wasy
de pfb-files in: localtexmf\fonts\type1\hoekwater\wasy
de afm-files in: localtexmf\fonts\afm\hoekwater\wasy
de pfm-files in: localtexmf\fonts\pfm\hoekwater\wasy
```
- Initialiseer MikTeX weer en het zou moeten werken.
- Natuurlijk kan dit ook in de texmf-boom, maar omdat dit aan de oorspronkelijke boom wordt toegevoegd zet ik het in de localtexmf-boom. Het is wat veel gedoe, maar wel overzichtelijk en dat maakt het zoeken en (evt) vervangen lekker eenvoudig.
- 
- Johan Vromans* **Afbreken met \raggedright**
- Misschien een FAQ-item, ik heb het mezelf in elk geval al vaak afgevraagd... Hoe kan ik, in geval van niet-uitgevulde regels (\raggedright) toch bereiken dat woorden worden afgebroken zodat de verschillen tussen de regellengtes kleiner worden? Bij voorkeur d.m.v. het instellen van een gewenste minimale regellengte.
- Piet van Oostrum* Kijk eens naar ragged2e.sty
- 
- Strijbos, Dr. R.C.* **Doorstrepen**
- Is er een eenvoudige manier in L<sup>A</sup>T<sub>E</sub>X om doorgestreepte tekst te krijgen, dwz een horizontale streep door het midden van de tekst(regel) om aan te geven dat die tekst uit het manuscript verwijderd moet worden?
- Piet van Oostrum* Zie het ulem of soul package (ulem.sty of soul.sty)
- ```
ftp://ftp.ntg.nl/pub/tex-archive/macros/latex/contrib/other/misc/ulem.sty
ftp://ftp.ntg.nl/pub/tex-archive/macros/latex/contrib/supported/soul/
```
-
- Eric Van Steenberg* **Nummering aanpassen van section**
- In een boek dat ik in L^AT_EX aan het verwerken ben moet ik de nummering van de sections telkens per chapter terug op nul kunnen beginnen (alsook het eerste chapter moet met het nummer 0 beginnen). Standaard begint deze nummering op 1. Hoe kan ik dit op 0 laten beginnen. Met \setcounter{chapter}{nummer} kan ik elk nummer wijzigen (zelfde geldt voor de section counter) maar krijg deze counters niet op nul.

Piet van Oostrum Na elk chapter / section commando:

```
\setcounter{chapter}{-1}
\setcounter{section}{-1}
```

Wybo Dekker **Raar fontprobleem latex/pdftex**

Voor een toepassing die a) op een kleine machine moet draaien en b) minimale pdf's moet produceren heb een tex-installatie met alleen times en mathpm fonts en alleen pdflatex. Ik heb als testfile:

```
\documentclass[pdftex]{article}
\usepackage{times,mathptm}
\begin{document}
 $x^y$ 
\end{document}
```

Het vreemde is dat pdflatex door de aanwezigheid van x^y toch cmfonts blijkt nodig te hebben en dus fout loopt, terwijl L^AT_EX dat probleem niet heeft. Ik krijg namelijk een foutmelding (zie onder), maar als ik die negeer krijg ik een pdf-file waar helemaal geen cm-fonts inzitten, en die er ook keurig als times uitziet. Wanneer ik echter de x^y door > vervang loopt acrobat reader ook nog fout.

Piet van Oostrum L^AT_EX hoeft niet te weten wat er in de fonts zit, die heeft alleen .tfm files nodig. De inhoud van de fonts wordt L^AT_EX door de dvi driver opgehaald. Voor times/mathptm worden virtuele fonts gebruikt (zptmcmrm o.a.) en L^AT_EX heeft dus alleen zptmcmrm.tfm e.d. nodig. zptmcmrm is opgebouwd uit een aantal fonts o.a. cmml10. Maar nogmaals: L^AT_EX weet dat niet want die kijkt niet in .vf files.

Pdflatex echter moet wel de inhoud van de fonts bekijken want die heeft – om zo te zeggen – zijn eigen dvi driver ingebouwd. Die moet dus de .vf bekijken om te zien wat de echte fonts erachter zijn, en dat is o.a. cmml10, dus die moet cmml10.tfm hebben. Op dat moment weet ie nog wel niet of je ooit een char uit cmml10 gaat gebruiken, maar de .vf file wordt kennelijk in zijn geheel bekeken en niet demand-driven.

Voor x^y heb je dat niet echt nodig, daarom gaat het bekijken van de pdf file wel goed. zptmcmrm.vf gebruikt voor een aantal tekens, waaronder ook '>' van cmml10. Waarom ze '>' niet uit Times of Symbol gehaald hebben weet ik niet, misschien vonden ze het niet wiskundig genoeg eruit zien. Dus als je die gebruikt wordt cmml10 echt opgenomen en komt Acrobat Reader natuurlijk in de problemen.

Suggestie: Laat in ieder geval de cmmath fonts tfm files op de machine staan (cmml10, cmex, cmsy, misschien ook in andere maten), en ook de pfb files voor die symbolen die eruit gehaald worden.

Of: gebruikt niet mathptm maar mathtime. Je kunt gratis vervangers van de basis mathtime fonts ophalen en met wat geknutsel installeren. Dit is deze maand op comp.text.tex besproken. Je krijgt dan een wat consistentere font, maar je hebt dan natuurlijk ook weer tfm en pfb files nodig dus dit levert geen ruimtebesparing op t.o.v. de eerste oplossing.

Of je moet zelf nieuwe virtuele fonts maken die de '>' e.d. symbolen wel uit Times en Symbol halen.

Stefan Goeman **Inhoudstafel**

Is er een handige manier om voor een bepaald hoofdstuk de `tocdepth` te veranderen? Blijkbaar kan je `\setcounter{tocdepth}{3}` alleen maar in de preamble gebruiken.

Ik zal het probleem een beetje duidelijker omschrijven. Sinds kort mogen wij aan de UG (Univ. Gent) onze doctoraten in het Engels schrijven. Indien we dit doen zijn we wel verplicht om een nederlandse samenvatting bij te voegen. Die nederlandse samenvatting stop ik momenteel in een appendix. Nu wil ik eigenlijk vermijden dat er in de inhoudstafel tweemaal hetzelfde komt. Eerst een gedetailleerde engelse inhoudstafel van al de hoofdstukken. Die inhoudstafel beschrijft dus ook de inhoud van de appendix en dat vind ik te gedetailleerd omdat dit ongeveer opnieuw de volledige inhoud overloopt. Ik wil dus eigenlijk iets in de stijl van:

```

1 Hoofdstuk 1:..... 1-1
  1.1 sect 1 .....
  1.2 sect 2 .....
    1.2.1 subsec .....
    1.2.2 subsec .....
2 Hoofdstuk 2 ..... 2-1
  2.1 sect 1 .....
    2.1.1 subsec .....
      2.1.1.1 subsubsec .....
      2.1.1.2 subsubsec .....
enz.
7 Hoofdstuk 7 ..... 7-1
  7.1 sec 1 .....
  7.2 sec 2 .....

A Nederlandse samenvatting ..... A-1
A.1 Hoofdstuk 1 ..... A-4
A.2 Hoofdstuk 2 ..... A-10
A.3 Hoofdstuk 3 ..... A-16
A.4 Hoofdstuk 4 ..... A-22
A.5 Hoofdstuk 5 ..... A-28
A.6 Hoofdstuk 6 ..... A-34
A.7 Hoofdstuk 7 ..... A-40
    
```

Piet van Oostrum Zet voor het betreffende `\chapter` commando:

```
\addtocontents{toc}{\setcounter{tocdepth}{1}}
```

Je moet overigens wel 3 keer latexen voor het effect doorgewerkt is.

Let op: in het geval dit hoofdstuk met `\include` ingevoegd wordt moet dit commando in de te includen file staan, niet in de hoofdfile, omdat het effect in het laatste geval een hoofdstuk (of beter gezegd: file) te laat optreedt. Dit is een L^AT_EX bug die erg moeilijk op te lossen is.

Als je `\usepackage{calc}` hebt, maak er dan van:

```
\addtocontents{toc}{\protect\setcounter{tocdepth}{1}}
```

Stefan Goeman Ik heb ondertussen al een betere (?) oplossing gevonden voor mijn inhoudstafel probleem. Door de *-vorm van de de commando's `subsection`, ... (dus `\subsection*`), `subsubsection*`, ... kan je ook vermijden dat die in de inhoudstafel verschijnen.

Dit lijkt mij een zeer handige manier, waarschijnlijk de beste, zo heb je ook geen last van de bug die Piet in een van de vorige mails beschrijft.

*Bert Frederiks***Gulden**

Hoe maak ik een mooie f voor onze nationale munteenheid?

*Piet van Oostrum*Dit advies kost `\emph{f}\,100,00`.*Taco Hoekwater*

De meeste 8-bit fonts hebben een 'mooi' karakter voor de gulden.

`\usepackage{textcomp}``\textflorin`*Frans Goddijn*

ik doe altijd fl wat toch een mooi ligatuurtje oplevert.

Wybo Dekker

Ik vind het allebei niks: Taco levert een rechtopstaande f die er net zo uitziet als een gewone f, bij Piet is de ruimte tussen f en 100 wel erg groot. Zelf gebruik ik altijd `\emph{f}100,00`. Ik heb nog even wat geëxperimenteerd om mezelf in die overtuiging te sterken:

```

\documentclass[12pt]{article}
\usepackage{textcomp}
\begin{document}\noindent
\begin{tabular}{lr}
\textflorin:          & \textflorin100,--\\
gewoon f:             & f100,--          \\
\emph:                & \emph{f}100,--  \\
\emph+small space:   & \emph{f}\,100,-- \\
\emph+italic correction: & \emph{f}\!/100,-- \\
\em:                  & {\em f}100,--   \\
\em+small space:     & {\em f}\,100,-- \\
\em+italic correction: & {\em f}\!/100,-- \\
\cline{2-2}
Totaal voor dit advies: & \emph{f}800,-- \\
\end{tabular}
\end{document}

```

*Ching Bon Lam***Minipages**

Ik wil graag twee minipage's naast elkaar hebben.

```

\begin{minipage}[b](7cm)
...
\end{minipage}
\hfill
\begin{minipage}[b](7cm)
...
\end{minipage}

```

dus iets van zo

```

*****
*      *      *      *
* tabel *      * tabel *
*      *      *      *
*      *      * tabel *
*      *      *      *

```

```
*           *           *****
*           *
*           *
*           *
*****
```

het enige probleem die ik tegenkom is dat de eerste regel van beide minipage's zich niet op dezelfde regel bevinden... ik worstel al een paar uur ermee en kan het antwoord maar niet vinden, hopelijk weten jullie het.

Paul Huygen In het volgende document wel:

```
\documentclass[11pt,a4wide]{artikel3}

\begin{document}
\mbox{
  \begin{minipage}[t]{4cm}
    \begin{tabular}[t]{cc}
      \textbf{links} & \textbf{rechts} \\
      1 & 10 \\
      2 & 20 \\
    \end{tabular}
  \end{minipage} \hfill
  \begin{minipage}[t]{4cm}
    \begin{tabular}[t]{cc}
      \textbf{links} & \textbf{rechts} \\
      1 & 10 \\
      2 & 20 \\
      3 & 30 \\
      4 & 40 \\
      5 & 50 \\
      6 & 60 \\
    \end{tabular}
  \end{minipage}
}
\end{document}
```

Eric Van Steenberghe **Paginanummer**

Hoe kan ik in L^AT_EX met `book.cls` het paginanummer onderdrukken voor een paar pagina's. De eerste pagina's mogen namelijk geen paginanummer tonen.

Edwin Drost `\thispagestyle{empty}` of `\pagestyle{empty}`

Maarten Wisse **Omgekeerde krul onder letters**

Ik ben bezig met het zetten van een zeventiende eeuwse latijnse tekst. Ik wil die zo getrouw mogelijk reproduceren. Daarin staan regelmatig letters met een krul eronder die precies de andere kant uitwijst dan de krul onder de franse c-cedille. Het gaat dan om een afkorting van combinaties als ae en oe. Ik heb pas gezien dat die omgekeerde krul wel in de standaardset van postscript fonts zit. Weet iemand een methode om die krul in L^AT_EX te verwerken?

Jörg Knappen `\usepackage[T1]{fontenc}`
`\k{a}` etc.

*Ramon Hanssen***Doorstrepen en weghalen**

Ik ben een draft van een manuscript aan het verbeteren. Over deze verbeteringen moet ik corresponderen met andere auteurs. Ik gebruik: `\usepackage{ulem}` om tekst door te strepen, met eigen commando `\newcommand{\doorgestreept}[1]{\xout{#1}}` Dat werkt prima. Wanneer de co-auteurs het eens zijn met mijn wijzigingen, wil ik op een makkelijke (en omkeerbare) manier de doorgestreepte text verwijderen uit het manuscript. Het liefst zou ik dus het commando `\doorgestreept` even in m'n preamble opnieuw willen definiëren, zodat de tekst tussen accolades verwijderd wordt na compilatie. Ik denk aan iets zoals

```
% \newcommand{\doorgestreept}[1]{\xout{#1}}
\newcommand{\doorgestreept}[1]{\verwijder{#1}}
```

Kent iemand een commando dat tekst tussen accolades verwijdert (dus niet uit de *.tex file, maar uit de dvi-file)?

Taco Hoekwater

```
\newcommand{\verwijder}[1]{} of direct: \newcommand{\doorgestreept}[1]{}
Maarten
```

Maarten

```
\makeatletter
\newcommand{\doorgestreept}[1]{\@bsphack\@esphack}
\makeatother
```

werkt wellicht nog iets beter.

*Gert Bartholomeeusen***CV**

Momenteel ben ik mijn cv-tje aan het schrijven. Echter ik weet niet goed hoe is dit op een deftige manier in L^AT_EX gedaan krijg. Ik wil iets in de volgende stijl

```
Gert Bartholomeeusen
curriculum vitae

Contact address: 35 Silver R..
Oxford ...
Uni...
...
Date of birth: May 3, 1974
Place of birth: Hoogstra...

EDUCATION
* DPhil, Civil Engineering, ...
* Master of Engineering, ...
* ...

WORK EXPERIENCE
* ...
* ...
* ...

....
```

Staat er puur toeval ergens iets op het net voor een cv in L^AT_EX? Uiteindelijk krijg ik de dingen wel geprint, maar het lijkt aaneen te hangen met haken en ogen.

Piet van Oostrum

Kijk eens naar `macros/latex209/contrib/resume`

Edwin Drost **Longtable/supertabular + tabularx**

Is er een mogelijkheid om automatisch de breedte van een kolom te bepalen en ook gebruik te maken van het automatisch splitsen van een tabel over twee pagina's? Ofwel kun je in subject genoemde packages combineren?

Piet van Oostrum Packages: `ltablex` of `ltxtable` (de laatste is van David Carlisle himself)

Edwin Drost Na een beetje prutsen heb ik de indruk gekregen dat `ltablex` het handigst werkt, aangezien je bij `ltxtable` de tabel vanuit je \LaTeX file moet exporteren of iets dergelijks. Wellicht heb ik een en ander verkeerd begrepen, een ander kan hier wellicht zijn voordeel mee doen.

TEXNiques

Toolbox: een syllabus

abstract

In deze toolbox wordt een voorbeeld besproken van een redelijk omvangrijk LATEX-document. Het nut van de verschillende packages komt aan de orde en er wordt tevens besproken hoe de packages worden gebruikt in het document zelf en welke opties daarvoor in de preamble moeten worden opgenomen.

keywords

LATEX, packages, praktijkvoorbeeld

Deze toolbox is weer gewoon in het Nederlands. Eigenlijk wel zo leesbaar en het schrijft nog sneller ook. Ook in een ander opzicht wijkt deze toolbox af van eerdere toolboxes. Ik beschrijf dit keer geen verzameling van op zichzelf staande trucs, maar loop gewoon eens door een document heen dat ik zelf onderhanden heb. Ik ben zelf altijd wel benieuwd hoe de documenten van anderen eruit zien en heb ervaren dat er een hoop LATEX-tips en -trucs zijn die je in geen handboek leest maar alleen ziet wanneer je eens een document van een ander onder ogen krijgt. Die persoon zelf heeft meestal niet eens in de gaten dat hij jou op dat moment wat bijbrengt. Hij (of zij natuurlijk, maar de TEX wereld bestaat voor een wel erg groot deel uit mannen) doet het altijd zo en kan zich niet voorstellen dat niet de hele wereld op die manier werkt.

Ten gevolge van de gekozen opzet is deze toolbox verre van 'self-contained'. Een aantal zaken stip ik alleen maar aan. Ik geef bij voorbeeld slechts aan wat *ik* in mijn document met een package doe. De door mij gebruikte functionaliteit vormt slechts een deel van de functionaliteit die het package biedt. Gelukkig is de documentatie van de packages zelf over het algemeen zeer goed. Deze kan de lezer bij voorbeeld vinden op de TEXLive CD. Onder de directory `/texmf/doc/latex` van deze CD is voor de meeste genoemde packages uitgebreide informatie in DVI- of postscript-formaat te vinden. Daarnaast is en blijft het natuurlijk mogelijk om voorbeelden te kopiëren en aan te passen zonder ze echt te snappen. Mijn ervaring is zelfs dat dit bij het gebruik van LATEX noodzakelijk is. Hoe alle LATEX-code aan elkaar gekoppeld is, is voor een gewone, welwillende gebruiker niet te begrijpen. Door wat code uit een package te halen en te experimenteren met kleine wijzigingen kunnen vaak echter de gewenste effecten zonder al te veel problemen gerealiseerd worden. Een voorbeeld van dit type aanpassing is de weergave van de literatuurlijst, zoals die later in deze toolbox wordt besproken.

Het document aan de hand waarvan ik dit verhaal opzet is een syllabus voor het vak Methoden en Technieken van Bedrijfswetenschappelijk Onderzoek. Het document is ongeveer 250 pagina's lang. De omvang van het hoofddocument is bijna 900 kilobyte.¹ Daarnaast worden voor het aanmaken van tabellen, figuren en de literatuurlijst nog vele andere bestanden gebruikt. Voor het bijhouden van verschillende versies van de syllabus gebruik ik RCS. De preamble van het document is meer dan 300 regels lang. Deze extreme omvang is overigens voor een groot deel het gevolg van het feit dat ik nooit de moeite heb genomen om een aantal zaken eens netjes tot een package te verwerken of tenminste via een input-commando op te vragen.

1. Met emacs valt met een document van deze omvang nog prima te werken. Wie werkt met een andere ASCII-editor geeft er wellicht de voorkeur aan om een document van deze omvang in een aantal subdocumenten op te splitsen.

Het begin van de preamble

In tegenstelling tot wat wellicht gebruikelijk is, begint het besproken bestand niet met het commando `\documentclass`, maar met de volgende if-constructie:

```
\newif\ifpdf
\ifx\pdfoutput\undefined
  \pdffalse
\else
  \pdfoutput=1
  \pdftrue
\fi
```

Hier wordt vastgesteld of ik de gewone TEX-executable, of pdfTEX gebruik, zodat ik later afhankelijk van het antwoord op deze vraag bij voorbeeld de juiste opties mee kan geven bij het laden van een package of de juiste plaatjes kan opvragen (terzijde zij opgemerkt dat de syllabus in zijn huidige vorm eigenlijk alleen onder een gewone TEX compileert). De reden om direct met deze if te beginnen en hem niet pas na het inladen van documentclass en packages op te nemen is dat sommige packages `\pdfoutput` ook definiëren als de gewone executable wordt gebruikt.

Documentclass en gebruikte packages

Met het volgende commando wordt de documentclass gespecificeerd en wordt een aantal packages geladen.

```
\ifpdf
  \documentclass
    [pdftex,dutch,bf,small,twoside,notitlepage,round,sort&compress]
    {syllabus}
\else
  \documentclass
    [dvips,dutch,bf,small,twoside,notitlepage,round,sort&compress]
    {syllabus}
\fi
\usepackage{amsmath,natbib,babel,tabularx,url,fancyvrb,textcomp}
\usepackage{caption2,wrisym,relsize,mflogo,lijstmac}
\usepackage{answers,makeidx}
%\usepackage[T1]{fontenc}
%\usepackage[bookmarks=true,bookmarksopen=true]{hyperref}
```

De documentclass die ik gebruik is `syllabus`; een eigen aanpassing van de documentclass `refart` die ik netjes hernoemd heb om te zorgen dat documenten die er vanuit gaan dat een ongewijzigde `refart` beschikbaar is gewoon blijven compileren.

Aan de documentclass geef ik een hele set opties mee. Het merendeel van de opties zou ik ook aan de packages mee kunnen geven. Ik vind het echter efficiënter om ze achter `\documentclass` op te nemen. Opties die bij `\documentclass` staan worden automatisch aan alle relevante packages doorgegeven. Dit scheelt niet alleen typewerk, maar maakt een document ook gemakkelijker onderhoudbaar: een wijziging hoeft nog maar op één plaats te worden aangebracht.

De meegegeven opties zelf zijn over het algemeen slechts voor één enkel package bestemd. Als eerste optie wordt, afhankelijk van de eerder gedefinieerde if, aan alle packages meegegeven of `pdftex` of `dvips` wordt gebruikt. Dit is bij voorbeeld van belang voor het

aanmaken van specials² voor afbeeldingen (iets waar het package `graphicx` zich mee bezig houdt) het aanmaken van specials voor hypertext-functies (iets waar `hyperref` zorg voor draagt). Doordat de optie bij `\documentclass` is meegegeven wordt deze automatisch doorgegeven aan alle relevante packages. Als ik bij voorbeeld het package `geometry` zou toevoegen om het pagina-formaat te wijzigen, zou deze ook automatisch de juiste optie meekrijgen.

De tweede optie (`dutch`) geeft aan dat we te maken hebben met een Nederlandstalig document. Met behulp van `babel` worden zo de juiste namen voor afbeeldingen, hoofdstukken enzovoorts geladen en in principe kan ook `natbib` deze informatie gebruiken om biografische verwijzingen en de literatuurlijst op correcte wijze weer te geven.

De opties `bf` en `small` worden allebei gebruikt door het package `caption2`. Het genoemde package dient om de weergave van bijschriften bij figuren en tabellen aan te passen. Ik geef er de voorkeur aan om de woorden ‘figuur 1’ vet te zetten en het hele bijschrift in een iets kleiner lettertype te zetten dan de rest van het document en dat is precies wat deze twee opties doen.

De optie `twoside` geeft aan dat het document dubbelzijdig geprint gaat worden (bij mijn weten heeft dit in dit document alleen invloed op de weergave van de kop- en voetteksten). Ook `notitlepage` zal bij de meeste lezers wel bekend zijn. Op deze manier wordt aangegeven dat TeX geen titelpagina moet maken. Deze optie is overigens wel een typisch voorbeeld van verkeerd toegepaste markup. De optie suggereert dat er geen titelpagina is. Deze is er wel degelijk, maar wordt in het document met de hand aangemaakt. Als TeX zelf nu ook nog een titelpagina toe zou voegen zouden er opeens twee zijn en dat is niet de bedoeling. Het zou natuurlijk veel netter zijn de weergave van de titelpagina in de documentclass `syllabus` zo aan te passen dat de automatisch aangemaakte titelpagina correct is.

De opties `round` en `sort&compress` worden allebei gebruikt door het package `natbib`. De combinatie van dit package en deze opties zorgt ervoor dat literatuurverwijzingen volgens het auteur-jaartal systeem worden weergegeven (dus in de vorm Jansen, 1999), dat de haakjes om literatuurverwijzingen rond zijn (standaard gebruikt `natbib` blokhaken) en dat verwijzingen gesorteerd en en waar mogelijk gecomprimeerde worden. Op deze manier wordt de verwijzing (Jansen, 1999c; Jansen 1999a; Jansen 1999b; Jansen 1998) automatisch vervangen door (Jansen, 1998, 1999a,b,c) hetgeen ruimte bespaart en wel zo leesbaar is.

Met een aantal `\usepackage`-regels worden vervolgens de packages zelf geladen. Ik ga ze weer even stuk voor stuk langs. Allereerst is er `amsmath`. Het `LATEX`-package van de American Mathematical Society. Dit package laadt een aantal commando’s die het ingeven van en verwijzen naar formules gemakkelijk maken. Met name voor formules die meerdere regels beslaan is dit van belang. Ook definieert dit package een aantal commando’s die het mogelijk maken formules netter te definiëren. Met standaard `LATEX` bestaat een formule veelal uit een set symbolen. Het `AMS`-package maakt het mogelijk om in plaats van naar de symbolen naar de betekenis van die symbolen te verwijzen en om gemakkelijk nieuwe functies te definiëren. De volgende formule

$$y_{ij} = \begin{cases} 0 & \text{if the expert did not mention the intended variable} \\ \frac{1}{q} & \text{if the expert mentioned } q \text{ variables, including the intended one} \\ 1 & \text{if the expert only mentioned the intended variable} \end{cases}$$

is ingegeven met de volgende standaard-`LATEX`-commando’s

2. `Special` is de TeX-term voor informatie die ‘letterlijk’ in het DVI-bestand wordt opgenomen en door de DVI-driver gebruikt kan worden voor het aansturen van bij voorbeeld de printer. Specials zijn enerzijds de kracht van TeX (zij zorgen voor uitbreidbaarheid) en anderzijds de zwakte (compatibiliteit is door de grote vrijheid bijna niet te waarborgen).

```

\begin{displaymath}
y_{ij} = \left\{
\begin{array}{l}
0 \text{ \mbox{if the expert did not mention the intended variable}} \\
\frac{1}{q} \text{ \mbox{if the expert mentioned } $q$ \text{ variables,} \\
\text{including the intended one}} \\
1 \text{ \mbox{if the expert only mentioned the intended variable}}
\end{array}
\right.
\end{displaymath}

```

Met behulp van de functionaliteit van `amsmath` kan de volgende, inhoudelijker, weergave worden gebruikt:

```

\begin{displaymath}
y_{ij} = \begin{cases}
0 \text{ \text{if the expert did not mention the intended variable}} \\
\frac{1}{q} \text{ \text{if the expert mentioned } $q$ \text{ variables,} \\
\text{including the intended one}} \\
1 \text{ \text{if the expert only mentioned the intended variable}}
\end{cases}
\end{displaymath}

```

Bovendien wordt enige nieuwe wiskundige functionaliteit ter beschikking gesteld (overigens, het hier niet geladen package `amsmath` zorgt dat een aantal extra symbolen zoals \mathbb{R} , \mathbb{C} via `\mathbb{R}`, `\mathbb{C}` kan worden gebruikt). Bijkomend voordeel van het gebruik van `amsmath` is dat in een aantal gevallen de opmaak van formules ‘verbetert’ (over smaak valt natuurlijk te twisten, maar het eindresultaat komt beter overeen met mijn verwachtingen en dat beschouw ik als een voordeel).

De packages `natbib` en `babel` zijn bij de opties al kort genoemd en zorgen voor respectievelijk extra functionaliteit bij het gebruik van citaties (zo komen commando’s als `\citeauthor` en `\citeyear` beschikbaar en kan in plaats van `\cite` gebruik gemaakt worden van `\citet` (voor verwijzingen in de lopende tekst) en `\citep` (voor verwijzingen tussen haakjes)) en taalgerelateerde aanpassingen.

Het package `tabularx` biedt een uitbreiding van de standaard `tabular`-omgeving van \LaTeX . Naast de `tabular`-omgeving, bestaat er nu ook een `tabularx`-omgeving. Aan deze omgeving kan een extra optie worden meegegeven waarin staat hoe breed de tabel moet worden. \LaTeX past de breedte van vooraf gespecificeerde kolommen dan automatisch aan teneinde de tabel de juiste breedte te geven. Om deze functie enigszins gemakkelijk te laten werken, worden later enige nieuwe kolomtypen gedefinieerd, die in een aparte paragraaf van deze toolbox samen met een voorbeeld besproken zullen worden.

Het volgende package luistert naar de naam `url` en definieert het commando `\url`. Dit commando is, het kan ook haast niet anders, uiterst geschikt voor de weergave van URL’s. Het zorgt dat de gebruikte tekens goed worden weergegeven (probeer deze (onzinnige) URL maar eens met de hand in te geven: `http://www.cwis_vu.nl/~maarten\lange\url\om\afbreken\te\demostreren&query=?`) en indien nodig op logische plaatsen worden afgebroken.

`fancyvrb` stelt extra functionaliteit voor `verbatim`-omgevingen beschikbaar. In de syllabus dient herhaaldelijk `spss`-syntax te worden opgenomen en dit package verschaft daarvoor de benodigde gereedschappen. Hier kom ik later nog op terug.

Dankzij `textcomp` heb ik de beschikking over een aantal extra tekentjes/symbolen zoals \textdegree (`\textdegree`), \textsterling (`\textsterling`), \textsection (`\textsection`) en \texttrademark (`\texttrademark`). Ook het commando `\textcircled` kan bruikbare diensten bewijzen. Bij dit alles dient overigens wel opgemerkt te worden dat niet alle symbolen even mooi zijn. De euro (\texteuro) is bij

voorbeeld ronduit lelijk en ook de Japanners komen er met hun ¥ (\textyen) wat bekaaid af. Zaken als \textcircled{R} (\textcircled{R}) lijken evenmin op wat ik zou verwachten³.

Het package caption2 dient, zoals reeds is gezegd, voor het aanpassen van de weergave van de bijschriften bij figuren. Het volgende package (wrisym) is voor de meeste lezers niet van belang. Het zorgt ervoor dat ik vanuit L^AT_EX de fonts met de wiskundige symbolen van het programma Mathematica kan gebruiken. Bovendien wordt de tekst automatisch uit times gezet. Een ‘poor man’s solution’ is \usepackage{times, mathptm}.

Het package relsize is typisch zo’n package dat functionaliteit biedt die eigenlijk in L^AT_EX zelf zou moeten zitten. Het zorgt ervoor dat er naast commando’s als \small en \large relatief werkende commando’s zoals \smaller en \larger beschikbaar komen. Deze commando’s kijken wat de huidige lettergrootte is en selecteren vervolgens een iets kleiner of groter lettertype. Handig voor tekst waarvan je van tevoren nog niet weet in wat voor grootte deze gezet gaat worden.

Een voorbeeld van het gebruik van het commando \smaller is te vinden in de volgende commandodefinitie. Hier wordt, later in mijn preamble, een commando \toets gedefinieerd. Met dit commando kan ik op een willekeurige plaats aangeven dat een toets (bij voorbeeld Ctrl-C) moet worden ingedrukt.⁴

```
\newcommand{\toets}[1]{\smaller\framebox{#1}}
```

Ook mflogo is een relatief eenvoudig package. Het stelt het metafont- en metapost-logo (METAFONT en METAPOST (\MF{} en \MP{})) beschikbaar. De lezer zij overigens gewaarschuwd: dit zijn twee extreem ‘gevaarlijke’ commando’s. Allereerst kennen oudere versies alleen het metafont-logo. Ten tweede gaat er relatief vaak iets mis met het vinden van deze fonts. Er zijn zowel versies in omloop waarin de fontnaam met hoofdletters wordt geschreven als versies waarin deze met kleine letters wordt geschreven. Als de gebruikte files niet allemaal dezelfde conventie hanteren kan de printer uiteindelijk het font niet vinden, met alle ellendige gevolgen van dien.

Het package lijstmac bevat wat zelf ontwikkelde macro’s voor het maken van lijsten (waarover in een volgende toolbox meer). Het package answers dient voor het aanmaken van opgaven. Het biedt de mogelijkheid om antwoorden en opgaven in het brondocument bij elkaar te houden en ze toch op verschillende plaatsen (in dit geval deels achterin de syllabus en deels in een aparte docentenhandleiding) te printen. Het gebruik van dit package komt later in deze toolbox nog uitvoeriger aan bod.

Voor het aanmaken van de index bij de syllabus gebruik ik het package makeidx. De overige commando’s die ik voor het aanmaken van de index gebruik worden later besproken.

De packages fontenc en hyperref worden beiden niet geladen. Zij werken alletwee niet samen met de overige packages. Via fontenc zou ik eigenlijk graag T_I-encoding willen gebruiken, omdat dan ook woorden met accenten adequaat worden afgebroken. Dit leidt echter tot een conflict met het wrisym package. Het aan brengen van hyperlinks etc. voor een elektronische versie via hyperref werkte altijd keurig, maar sinds ik mijn T_EX heb geupgrade naar versie 5c van T_EXLive is dat verleden tijd.

Het laatste te laden package wordt op de volgende manier opgevraagd.

```
\ifpdf
\usepackage[pdftex]{graphicx}
```

3. *Produktienoot.* De euro en de cirkel voor omcirkelde letters bleken niet aanwezig in mijn versie van textcomp/Times. Deze twee, en de yen heb ik voor de MAPS gezet uit Computer Modern. Voor de cirkel kon ik gebruik maken van aer, een emulatie van het ec font met behulp van cm, dat in Type1-formaat beschikbaar is, en virtual fonts, maar de euro en de yen zijn de enige bitmapped letters uit de hele MAPS. Het marvosym pakket, beschikbaar in de teTeX distributie, heeft een aantal euro symbolen in Type1 formaat: €, € en €. De yen is aanwezig in de meeste commerciële fonts, b.v. voor Times ¥ [SK].

4. Ctrl-C werkt ook in voetnoten en gebruikt dan een navenant kleiner lettertype.


```

\graphicspath{{pdf/}}
\DeclareGraphicsExtensions{.pdf}
\DeclareGraphicsRule{*}{mps}{*}{}
\else
\usepackage[dvips]{graphicx}
\graphicspath{{afb/}}
\DeclareGraphicsExtensions{.eps}
\fi

```

Het zo ingeladen `graphicx` package biedt, zoals de naam al aangeeft wat grafische functionaliteit. Dit package gebruik ik onder andere om figuren in mijn document op te nemen. Daarnaast gebruik ik het om extreem grote tabellen integraal te verkleinen en eventueel te roteren, zodat zij nog op één pagina passen. Dit is met name van belang voor het weergeven van bij voorbeeld een z -tabel (voor de normale verdeling). Daar het `graphicx`-package gebruik maakt van *specials* laat ik de vraag met welke opties het geladen wordt afhangen van de vraag of ik met pdf_T_E_X werk of met een gewone _T_E_X (en dan uiteindelijk `dvips` wil gebruiken). Strikt genomen is dit door het gebruik van de globale optie overigens overbodig. Wat niet overbodig is, is het afhankelijk van de gebruikte software instellen van paden voor de vindplaats van afbeeldingen, grafische extensies en regels voor de verwerking van afbeeldingen.

Het aanmaken van de literatuurlijst

Een andersoortig ‘probleempje’ dat ik oplos in de preamble is dat de literatuurlijst standaard de verkeerde titel heeft (ik geloof bibliografie, of referenties of zoiets), terwijl ik wil dat daar gewoon ‘Literatuur’ boven komt staan. Door het herdefiniëren van `\bibsection` wordt dit probleem opgelost. Tegelijkertijd zorg ik ervoor dat de literatuurlijst netjes in de inhoudsopgave wordt opgenomen. Dit alles gebeurt met onderstaande commando’s.

```

% Literatuurlijst hernoemen en in inhoudsopgave opnemen
\makeatletter
\renewcommand\bibsection{\def\refname{Literatuur}}%
\chapter*{\refname
  \@mkboth{\MakeUppercase{\refname}}}%
  {\MakeUppercase{\refname}}%
  \addcontentsline{toc}{chapter}{\refname}}%
}%
\makeatother

```

Wat de lezer aan deze commando’s wellicht opvalt is het gebruik van `\makeatletter` en `\makeatother`. In commando-namen mag normaalgesproken het apestaartje (`@`, oftewel `at-sign`); vandaar de naam `\makeatother`) niet voorkomen. In het hier weergegeven commando heb ik echter het apestaartje wel nodig. Met het commando `\makeatletter` vertel ik _T_E_X dat het net moet doen of het apestaartje een gewone letter is. Na de definitie van het commando maak ik deze zogenaamde ‘catcode-wijziging’ weer ongedaan.

Het gebruik van `natbib` maakt het mogelijk de layout van de literatuurlijst zonder al te veel problemen aan te passen. Ik kies een iets kleiner lettertype en verklein tevens de afstand tussen de individuele titels.

```

\def\bibfont{\small}
\def\bibsep{\smallskipamount}

```

De literatuurlijst zelf wordt aangemaakt met behulp van het volgende commando.

```
{%\citeindexfalse
```

```
\bibliography{string,data,ik,crossref,tmp,mt}%\citeindextrue
}
\bibliographystyle{my-nl}
```

Achter de procent-tekens staan twee commando's die ik nu niet gebruik. Als ik ook een auteursindex aan zou maken, zou ik de procent-tekens weghalen om te voorkomen dat de literatuurlijst zelf in de auteursindex wordt opgenomen. Het commando `\bibliography` zorgt dat de literatuurlijst (die op de gebruikelijke manier met behulp van `bibTEX` is aangemaakt) wordt opgenomen. De parameters van dit commando geven aan welke `bib`-files `bibTEX` moet lezen. De parameter van `\bibliographystyle` geeft aan welk stijlbestand voor het opmaken van de literatuurlijst moet worden gebruikt (in dit geval een zelf-gemaakte Nederlandstalige versie met ondersteuning voor het auteur-jaartal systeem).

Een kleine opmerking over de gebruikte bestanden is nog op zijn plaats. Het eerste bestand (`string.bib`) bevat een serie afkortingen die ik in de andere bestanden gebruik. Deze afkortingen (in totaal zo'n 200) hebben de volgende vorm.

```
@STRING ( AOS = "Accounting, Organizations and Society" )
```

Het bestand `data.bib` bevat het merendeel van de literatuurverwijzingen. In dit bestand worden de eerder gedefinieerde afkortingen gebruikt. Dit scheelt niet alleen typewerk, maar zorgt er ook voor dat ik de namen van tijdschriften consequent spel en gemakkelijk naar een afgekort formaat (Acc., Org. & Soc.) over kan stappen.

```
@ARTICLE{Abernethy95a,
  author = {Abernethy, Margaret A. and Stoelwinder,
            Johannes U.},
  title = {The role of professional control in the
            management of complex organizations},
  year = 1995,
  journal = AOS,
  volume = 20,
  pages = {1-17},
  number = 1
}
```

De bestanden `ik.bib`, `tmp.bib` en `mt.bib` bevatten eveneens gegevens. Het bestand `crossref.bib` bevat de titels van artikelenbundels en dat soort zaken. Het is voor het gebruik van `bibTEX` noodzakelijk dat een artikel uit een bundel eerder in de `bib`-file staat dan de bundel zelf (hetgeen tot conflicten leidt met mijn wens mijn gegevensbestanden te sorteren). Een andere mogelijkheid is het opnemen van de bundels in een file die later geladen wordt dan de file met de artikelen. Dat is de optie die ik hier kies. De artikelen staan in `data.bib` en `ik.bib`, de bundels in `crossref.bib`.

Instellingen voor tabularx

Eén van de packages die eerder geladen is, is `tabularx`. Zoals gezegd biedt dit package uitbreidingen op de standaard `tabular`-omgeving van `LATEX`. De belangrijkste uitbreiding is dat aan een tabel een breedte mee kan worden gegeven en dat vooraf geselecteerde kolommen vervolgens automatisch breder of smaller worden gemaakt om zo de tabel de gewenste breedte te geven. Het package is zonder verdere instellingen prima te gebruiken, maar ikzelf geef er de voorkeur aan een iets 'toegankelijker' interface te definiëren. Hiertoe maak ik de volgende nieuwe kolomtypen aan:

```
% Nieuwe kolommen voor tabularx en array
\newcolumntype{L}{>{\raggedright\arraybackslash}X}
```

```
\newcolumntype{R}{>\raggedleft\arraybackslash}X}
\newcolumntype{C}{>\centering\arraybackslash}X}
```

De kolomtypen L, R en C vormen nu het ‘rekbare’ equivalent van de traditionele kolomtypen l, r en c. Nu kan ik in mijn tekst bij voorbeeld de volgende commando’s gebruiken:

```
\begin{tabularx}{0.67\textwidth}{Lrr}
Enige tekst&tekst&tekst\\
Enige tekst&tekst&tekst\\
\end{tabularx}
```

Om een tabel aan te maken die 0.67 keer zo breed is als de lopende tekst. De tweede en derde kolom worden gewoon rechts uitgelijnd. De eerste kolom wordt links uitgelijnd en net zo lang opgerekt tot de tabel breed genoeg is. Indien een kolom te lang is, wordt deze over meerdere regels verdeeld:

```
\begin{tabularx}{0.67\textwidth}{L|r|r}
Enige tekst en een boel tekst,
en een boel tekst, en een boel tekst,
en een boel tekst, en een boel tekst,
en een boel tekst, en een boel tekst,
en een boel tekst.&tekst&tekst\\
Enige tekst&tekst&tekst\\
\end{tabularx}
```

Leidt tot de volgende tabel:

Enige tekst en een boel tekst, en een boel tekst, en een boel tekst, en een boel tekst, en een boel tekst, en een boel tekst,	tekst	tekst
Enige tekst	tekst	tekst

Vanzelfsprekend is het mogelijk om meerdere ‘rekbare’ kolommen tegelijkertijd te gebruiken en om nog extra soorten kolommen te specificeren (bij voorbeeld om te zorgen dat de éne kolom twee keer zo snel groeit als de ander).

Gebruik van answers

Zoals reeds eerder aangegeven gebruik ik het package `answers` voor het weergeven van mijn opgaven, het maken van een set uitwerkingen en hints voor studenten en het aanmaken van het antwoordboekje. Uiteindelijk ziet een opgave er in mijn document zo uit:

```
\begin{opgave}
Hoeveel is  $1+1$ ?
\begin{hint}
Gebruik je rekenmachine!
\end{hint}
\begin{uitwerking}
2
\end{uitwerking}
\end{opgave}
```

De hele opgave staat in een `opgave`-omgeving. De hint (die in de syllabus zelf terecht moet komen) staat in een `hint`-omgeving. De uitwerking, die alleen voor docenten is bestemd, staat in de `uitwerking`-omgeving. Van sommige opgaven komt het antwoord ook in de syllabus terecht (tussen de hints), deze staan in een `antwoord`-omgeving.

Het opnemen van de antwoorden in de syllabus gaat als volgt. Het package `answer` plaatst de uitwerkingen in hulpfiles. Ik gebruik er twee. Eén voor docenten en één voor studenten. Voordat ik de hulpfiles in de bijlage kan opnemen, moet ik ze eerst sluiten met `\Closesolutionfile`. Vervolgens laad ik de studentenfile met behulp van het commando `\input`.⁵

```
\Closesolutionfile{docent}
\Closesolutionfile{student}
\chapter{Antwoorden en hints}
\label{cha:antwoorden}
\input{student}
```

Voordat `answer` werkt zoals hierboven beschreven staat, moet er wel het één en ander gebeuren. Allereerst wil ik mijn opgaven nummeren. In de commando's komen apestaartjes voor, dus ik begin met `\makeatother`. Vervolgens geef ik aan dat ik een nieuwe counter wil definiëren (die luistert naar de naam 'opgavenummer') en dat de opgavenummers ieder hoofdstuk opnieuw bij 1 moeten beginnen. De laatste hieronder gepresenteerde regel zorgt dat het opgavenummer wordt voorafgegaan door het hoofdstuknummer (anders maakt de helft van mijn practicumdeelnemers weer de opgaven uit een verkeerd hoofdstuk), waarna het nummer van de opgave zelf als arabisch getal (dat is een gewoon getal en geen romeins cijfer of een letter of zo) wordt weergegeven:

```
% Setup voor answers
\makeatletter
\newcounter{opgavenummer}[chapter]
\renewcommand{\theopgavenummer}{\thechapter.\@arabic\c@opgavenummer}
\makeatother
```

Dan moet ik de omgeving opgave nog definiëren. Aan het begin van deze omgeving wordt de opgave-teller automatisch verhoogd met één. Ik gebruik het commando `\refstepcounter` om te zorgen dat ik ook naar opgaven kan verwijzen. Iedere opgave begint een nieuwe paragraaf (`\par`) en aan het begin van de opgave wordt zonder in te springen (`\noindent`) in vet het woord 'opgave' en het nummer van de opgave weergegeven. Opgaves eindigen met een rechts uitgelijnd vierkant.⁶ Het rechts uitlijnen wordt bereikt door een oneindig rekbaar `\hspace` op te nemen.

```
\newenvironment{opgave}{%
\par\refstepcounter{opgavenummer}
\noindent\textbf{Opgave \theopgavenummer:}}
{\nobreak\hspace*{\stretch{1}}\nobreak\EmptySquare\par}
```

Nu moet ik nog zorgen dat de uitwerkingen, hints en antwoorden in de goede hulpfile terecht komen. Uitwerkingen komen in een bestand `docent.tex`, de hints en uitwerkingen moeten op volgorde in een bestand `student.tex` worden opgenomen. Als ik de hints en de uitwerkingen apart zou willen weergeven, zou ik voor beiden een apart hulpbestand moeten gebruiken.

```
\Newassociation{uitwerking}{Uitwerking}{docent}
\Newassociation{hint}{Hint}{student}
\Newassociation{antwoord}{Antwoord}{student}
```

De bijbehorende labels moeten nog zo worden gedefinieerd dat de uitwerkingen er een beetje fatsoenlijk uitzien:

5. Het uitwerkingenboekje is een aparte TeX-file, die niets anders doet dan de docentenfile opvragen binnen een document-omgeving.

6. Dit vierkant is afkomstig uit de symbolenset van Mathematica en dus niet standaard beschikbaar.

```

\renewcommand{\Uitwerkinglabel}[1]{\textbf{Uitwerking voor opgave #1:}}
\renewcommand{\Hintlabel}[1]{\textbf{Hint voor opgave #1:}}
\renewcommand{\Antwoordlabel}[1]{\textbf{Antwoord voor opgave #1:}}

```

Door middel van een ‘truc’ zorg ik er vervolgens voor dat het nummer van de juiste opgave, telkens wordt meegegeven aan de hierboven gedefinieerde kopjes van de opgaves. Overigens heb ik sterk het idee dat dit efficiënter moet kunnen via één of andere optie van het gebruikte package. Voorlopig ben ik echter al heel blij dat de boel het doet.

```

\def\nummer{\theopgavenummer}
\let\Currentlabel\nummer

```

Daarna hoef ik alleen de hulpfiles nog maar te openen, zodat de gegevens daadwerkelijk naar de hulpfile weggeschreven kunnen worden.

```

\Opensolutionfile{docent}
\Opensolutionfile{student}

```

Gebruik van fancyvrb

Het package fancyvrb gebruik ik voor het weergeven van spss-code. Voordat ik dit package gebruik, herdefinieer ik eerst het uitroepteken. Het uitroepteken komt in spss-syntax vrijwel niet voor en wil ik gebruiken als commando om cursieve tekst te selecteren voor delen van de syntax. Het eerste wat ik doe is het opslaan van de oorspronkelijke betekenis van het uitroepteken in het commando `\ExclamationPoint`. Vervolgens maak ik het uitroepteken actief. Het uitroepteken is nu geen uitroepteken meer, maar een commando. Tot slot definieer ik het commando uitroepteken en geef het standaard dezelfde betekenis als het zojuist gedefinieerde commando `\ExclamationPoint`.

```

\def\ExclamationPoint{\char'!}

\catcode'\!=\active
\def!\{\ExclamationPoint{}}

```

De spss-code moet in een omgeving genaamd SPSS terecht komen. Om deze omgeving te definiëren gebruik ik het commando `\DefineVerbatimEnvironment` zoals dat hieronder is weergegeven. Achter frame geef ik aan dat boven en onder de syntax een lijn moet komen, het lettertype moet redelijk klein zijn en ik herdefinieer het zojuist actief gemaakte uitroepteken. Het uitroepteken is nu geen uitroepteken meer, maar het commando `\em`. Verder geef ik aan dat de syntax bij elkaar moet worden gehouden op een enkele pagina (dit werkt niet optimaal, de syntax blijft inderdaad keurig bij elkaar, maar de lijn boven/onder de syntax wil nog wel eens op de verkeerde pagina terechtkomen). Tot slot specificeer ik de afstand tussen syntax en de lijnen.

```

\DefineVerbatimEnvironment{SPSS}{Verbatim}{%
  frame=lines,
  fontsize=\small,
  defineactive=\def!\{\em},
  samepage=true,
  framesep=0.7\fbboxsep}

```

Nu heb ik een omgeving waarin ik de volgende truc kan uithalen:

```

\begin{SPSS}
  tekst !cursieve tekst! rechte tekst !cursief!
  En alle andere vervelende letters werken nog:
  ~@#$$%^&*()_+={}[ ]\
\end{SPSS}

```

```
tekst cursieve tekst rechte tekst cursief
En alle andere vervelende letters werken nog:
~@#$$%^&*()_+={}[|\
```

In de praktijk ziet een stukje SPSS-syntax er dan als volgt uit (`\cindex` zorgt ervoor dat het commando in de index wordt opgenomen).

```
\cindex{regression}
\begin{SPSS}
  regression
    /variables = {!variabelen!|(collect)|all}
    /descriptives = {defaults|mean|stddec|corr|cov|variance|
                    all|none}
    /missing = {listwise|pairwise}
    /statistics = {defaults|r|coeff|anova|outs|all}
    /dependent = !variabelen!
    /method = enter !variabelen!
    /residuals
    /casewise
    /scatterplot = !variabele!
    /partialplot = !variabele!
    /save = !tempvar!(!newvar!).
\end{SPSS}
```

```
regression
  /variables = {variabelen|(collect)|all}
  /descriptives = {defaults|mean|stddec|corr|cov|variance|
                  all|none}
  /missing = {listwise|pairwise}
  /statistics = {defaults|r|coeff|anova|outs|all}
  /dependent = variabelen
  /method = enter variabelen
  /residuals
  /casewise
  /scatterplot = variabele
  /partialplot = variabele
  /save = tempvar(newvar).
```

Het aanmaken van de index

Om een index aan te maken, moet ik natuurlijk eerst even zorgen dat de benodigde hulpbestanden automatisch worden gegenereerd. Dit bereik ik door het volgende commando in de preamble op te nemen.

```
\makeindex
```

Alleen bovenstaand commando zou al voldoende zijn voor het genereren van een bruikbare index (in de tekst moet natuurlijk nog wel aan worden gegeven wat er geïndexeerd dient te worden). In de praktijk heb ik echter de gewoonte nog een aantal andere zaken in de preamble op te nemen. Zo staat er ergens halverwege—het zal iedere lezer in de tussentijd duidelijk zijn dat mijn preamble één ongeordende chaos is—het volgende commando.

```
\newcommand{\cindex}[1]{\index{#1@\texttt{#1}}}
```

Betekenis en nut zijn wellicht niet direct duidelijk, maar niet moeilijk te begrijpen. Ik wil in de index onder andere spss-commando's opnemen. Deze moeten daar in typemachine-schrift worden weergegeven. Als ik daarvoor zonder voorzorgsmaatregelen het `\index`-commando zou gebruiken, zou mijn index niet langer goed gesorteerd worden. Gelukkig voorziet het `index`-commando in de `@`-functionaliteit. De index wordt gesorteerd op hetgeen voor de apostrof staat en weergegeven als wat er achter staat. Dit brengt echter een hoop onnodig typewerk met zich mee, vandaar dat ik bovenstaand commando `\cindex` aanmaak, dat slechts één parameter (de in typemachineschrift te verschijnen index-entry, oftewel het spss-commando) meekrijgt en zorgt dat deze goed gesorteerd en juist weergegeven in de index verschijnt. Een voorbeeld van het gebruik van dit commando staat bij de regressie-syntax op pagina 66.

Niet alle index-entries worden in de lopende tekst aangemaakt. Om de index bruikbaar te maken zijn ook verwijzingen van groot belang. Aan het einde van mijn preamble staat een hele set commando's die er voor zorgen dat die verwijzingen in de index worden opgenomen. Zij hebben allemaal de volgende vorm.

```
\index{degree of freedom|see{vrijheidsgraad}}
```

Die zou ik eigenlijk eens in een apart hulpbestand moeten plaatsen. Dat maakt de boel weer wat leesbaarder. Nog mooier zou het zijn om een commando `\indexverwijzing` aan te maken. Dat zou op de volgende manier moeten gebeuren.

```
\newcommand{\indexverwijzing}[2]{\index{#1|see{#2}}}
```

Aan het 'einde' van de syllabus moet de index natuurlijk nog geprint worden. Dit gebeurt door het volgende commando op te nemen.

```
{\printindex}
```

Dat vereist overigens wel dat er een index-bestand aanwezig is. L^AT_EX maakt slechts een hulpbestand aan, dat vervolgens door een extern programma (meestal `makeindex` geheten) gesorteerd moet worden. Normaalgesproken is dit een kwestie van de namen van de commando's intypen. Bij de syllabus treedt er echter een complicatie op. Ik heb het uitroepteken actief gemaakt en standaard dient het uitroepteken voor verwijzingen van verschillende niveaus (naast een index-entry factoranalyse, is er bij voorbeeld een entry, factoranalyse; vooronderstellingen en een entry factoranalyse; interpretatie. Normaalgesproken worden de niveaus in het `index`commando als volgt ingegeven:

```
\index{factoranalyse!vooronderstellingen}
```

Met het actief gemaakte uitroepteken werkt dit niet meer. In plaats daarvan gebruik ik de punt-komma.

```
\index{factoranalyse;vooronderstellingen}
```

Dit moet echter nog wel even aan `makeindex` worden verteld, anders loopt de boel in het honderd. Om `makeindex` de niveaus juist te laten verwerken, maak ik een bestand `syllabus.ist` aan met de volgende inhoud:

```
level ';' 
```

Door nu `makeindex` als volgt aan te roepen, wordt de index gegenereerd (de `c` dient om spaties te negeren).

```
makeindex -c -s syllabus syllabus
```

Wat extra commando's

Tussen alle commando's die direct aan een package gerelateerd zijn, staan in mijn preamble nogal wat, eigenlijk vrij willekeurige extra commando's, die niet veel meer doen dan mijn leven vergemakkelijken. Het eerste commando dient voor het opnemen van elementen van het rooster in de studiewijzer. Iedere week worden er één of meer onderwerpen besproken die over het algemeen overeenkomen met één of meer onderwerpen uit het boek van Hair en/of de hier besproken syllabus. Het onderstaande commando definieert een nieuw commando `\rooster` met 5 parameters: het weeknummer, de datum, het hoofdstuk uit Hair, het hoofdstuk uit de syllabus en een nadere omschrijving van het onderwerp. Door dit commando aan te roepen zorg ik ervoor dat de beschrijvingen van alle weken in mijn studiewijzer er consistent uitzien—en het bespaart ook nog eens typewerk.

```
% Collegerooster
\newcommand{\rooster}[5]{%
  \noindent\textbf{Week #1}
  \begin{description}
    \item [Datum] #2
    \item [Hair] Hoofdstuk #3
    \item [Syllabus] Hoofdstuk #4
    \item [Onderwerpen] #5
  \end{description}}
```

In de studiewijzer, gebruik ik dit commando als volgt:

```
\rooster
{5}
{28/2/2000}
{2}
{3 en 4}
{Het eerste uur wordt het onderwerp gegevensverzameling verder
uitgebreid. Er wordt ingegaan op case studies en het coderen van
gegevens. Het tweede uur wordt besteed aan verkennen en presenteren
van gegevens, het gebruik van tabellen en figuren wordt
overgeslagen, dit kan je zelf gemakkelijk bestuderen. Outliers,
normaliteit en het nut van gegevenstransformaties komen wel aan bod}
```

Op een vergelijkbare wijze worden ook commando's aangemaakt voor speciale lijsten etc. In sommige gevallen zijn de commando's iets complexer dan bovenstaand voorbeeld, omdat bij voorbeeld de kantlijnen worden aangepast of er zaken worden gerooteerd. Het principe is echter gelijk en ik zal dan ook niet al deze voorbeelden bespreken.

Een volgende cosmetische aanpassing is nog kleiner: ik wil dat er bij de voetnoten geen streep verschijnt. Dit effect bereik ik door het commando `\footnoterule`, waarmee de streep wordt gezet, te herdefiniëren. Door het commando `\renewcommand` zoals hieronder weergegeven wordt het bestaande `\footnoterule`-commando vervangen door `\relax`, oftewel: doe niets.

```
% Geen streep bij voetnoten
\renewcommand\footnoterule{\relax}
```

Een actie die de meeste lezers wel bekend voor zal komen is het aanmaken van een nieuwe theorem-environment. Het zal geen verbazing wekken dat ik in een syllabus over onderzoeksmethoden en -technieken af en toe een hypothese nodig heb. Dit wordt hieronder gedefinieerd. In dit geval heet de environment `hypo` en de starttekst is `Hypothese`.

```
\newtheorem{hypo}{Hypothese}
```


Ik hou ervan om hoofdstukken te beginnen met een kort citaat. Dit citaat moet wat smaller zijn dan de tekst, cursief worden weergegeven en worden gevolgd door de naam van de auteur en een jaartal. Zo ongeveer als hieronder, maar dan (in het geval van de syllabus) ook nog met paginanummer.

*Het rare van die wiskunde is dat het uiteindelijk toch altijd nog
ergens op wordt toegepast.* Hugo Brandt Cortius (1997)

Het volgende commando bereikt dit.

```
% Citaten (heeft op deze wijze wel natbib/plainnat nodig)
\newlength{\half}
\half=0.3\textwidth
\newenvironment{citaat}[2]{%
  \def\auteur{#1}\def\plaats{#2}%
  \list{}{\rightmargin\nomargin\leftmargin\half}
  \item[]\em\small}
  {\hspace*\stretch{3}}\nolinebreak[2]\hspace*\stretch{3}}%
  \textnormal{%
    \mbox{\citeauthor{\auteur}~(\citeyear{\auteur},~\plaats)}}%
  \endlist\medskip\ignorespaces}
```

Eerst definieer ik een nieuwe lengte, die ik de naam `\half` geef (kennelijk wilde ik in eerste instantie citaten half zo breed maken als de rest van de tekst, maar, zoals op de volgende regel te zien is, ben ik daarop teruggekomen, ze worden 0.7 keer zo breed; een typisch voorbeeld van hoe (mijn) macro's na verloop van tijd volledig onleesbaar worden). Na het definiëren van de lengte wordt een omgeving gedefinieerd. Deze omgeving heeft twee parameters: de sleutel van het citaat in mijn literatuurdatabase (noodzakelijk voor het `\cite`-commando) en de plaats (meestal een paginanummer) waar het citaat gevonden kan worden. Zoals de meeste \LaTeX -gebruikers zullen weten, bevat de definitie van een omgeving (zoals er hier één wordt aangemaakt, het citaat komt immers tussen `\begin{citaat}` en `\end{citaat}` te staan) twee parameters: de commando's die aan het begin van de omgeving moeten worden uitgevoerd en de commando's die aan het einde moeten worden uitgevoerd. Er treedt bij dit commando echter een complicatie op. De extra argumenten van een omgeving (in dit geval de sleutel voor de database en het paginanummer) zijn slechts aan het begin van de omgeving, en niet aan het einde beschikbaar. Het eerste wat ik dan ook doe is deze eerste en tweede parameter opslaan in de commando's `\auteur` en `\plaats`. Hier gebruik ik het commando `\def` voor. Ik geloof dat dit niet geheel volgens de regels van goed \LaTeX -gebruik is, maar het is de gemakkelijkste mij bekende manier om te zorgen dat een commando dat nog niet bestaat wordt aangemaakt en een reeds bestaand commando rücksichtslos wordt overschreven, zodat bij een tweede aanroep van de citaat-omgeving de gegevens van het eerste citaat automatisch gewist worden.⁷

Na het opslaan van de parameters wordt de layout van de omgeving gespecificeerd. De gebruikelijke, maar weinig intuïtieve manier om dit in \LaTeX te doen is het gebruik van een `\list`. In dit geval een erg eenvoudige. Alleen de linker- en rechtermarge hoeven maar gedefinieerd te worden. Vervolgens dient er nog een `\item`-commando gegeven te worden (anders gaat \LaTeX klagen over een lijst zonder items) en daarna wordt het lettertype (cursief en klein) geselecteerd.⁸

Aan het einde van het citaat dient de naam van de auteur geplaatst te worden, samen met jaartal en paginanummer. Ik kies ervoor om deze gegevens allemaal rechttop (in romein)

7. Wellicht was het gebruik van `\bgroup` en `\egroup` eleganter geweest.

8. In mijn geval is `\nomargin` al gedefinieerd in de gebruikte documentclass. Dit is niet standaard het geval.

te zetten en ze op één regel bij elkaar te houden. Indien het citaat op de laatste regel voldoende ruimte overlaat voor de ‘aftiteling’ wordt deze rechts uitgelijnd achter de laatste regel van het citaat geplaatst. Is er te weinig ruimte, dan komt de hele regel op een nieuwe regel. Dit is, samen met het toevoegen van wat wit aan het einde van het totale citaat, ongeveer alles wat de laatste regels van het commando doen, maar deze exercitie vraagt nogal wat commando’s.

De regel:

```
{\hspace*{\stretch{3}}\nolinebreak[2]\hspace*{\stretch{3}}}
```

zorgt ervoor dat indien er genoeg ruimte is, de volgende ‘aftiteling’ direct achter de laatste regel van het citaat komt. Indien dit niet het geval is, komt de ‘aftiteling’ op een aparte regel, zoals hieronder.

*Validity refers to the capacity of a
test to tell us what we already know.*
George Kelly

De truc werkt als volgt. In bovenstaande regel staan twee oneindig rekbaar stukken wit. Deze zorgen er met zijn tweeën danwel voor dat de aftiteling die op dezelfde regel blijft staan rechts wordt uitgevuld, danwel dat de laatste regel van het citaat netjes wordt opgevuld met wit en op de volgende regel de aftiteling links met wit wordt opgevuld (zodat de aftiteling rechts uitgevuld verschijnt). Het commando `\nolinebreak` zorgt ervoor dat tussen beide stukken oneindig rekbaar wit bij voorkeur geen nieuwe regel wordt begonnen, zodat de aftiteling als het even mogelijk is direct achter het citaat terecht komt.

Op de volgende en laatste regel van de commandodefinitie wordt het standaard lettertype weer gekozen, zorgt de `\mbox` ervoor dat zijn inhoud op één enkele regel blijft staan (ook als die regel te kort is), halen `\citeauthor` en `\citeyear` respectievelijk auteur en publicatiejaar uit de literatuurlijst en wordt vervolgens de lijst afgesloten. De gebruikte `\cite`-commando’s vereisen overigens wel dat `natbib` wordt gebruikt in combinatie met een adequate bibliography-style.

```
\textnormal{%  
  \mbox{\citeauthor{\auteur}~(\citeyear{\auteur},~\plaats)}}%  
\endlist\medskip\ignorespaces}
```

Na deze citaat-exercitie volgt een ander commando uit de categorie ‘restpost’ dat hieronder wordt besproken. Op een gegeven moment heb ik er in de syllabus de behoefte een spatie weer te geven als `┐`. In lopende tekst zou dit geen probleem zijn, maar in dit geval vormt de spatie het argument voor een ander commando. Het gebruik van `\verb*| |` is derhalve niet mogelijk. Om dit probleem op te lossen maak ik een box genaamd spatie aan en in deze box plaats ik de `┐`. De box met de naam spatie kan ik vervolgens zonder problemen als argument van een ander commando gebruiken. Overigens is dit een typisch voorbeeld van een niet-elegante, maar wel werkende oplossing.

```
\newbox\spatie% Spatie bevat een verbatim spatie  
\setbox\spatie=\hbox{\verb*| |}
```

Als parameter van het commando `\hyphenation` volgt een aantal woorden dat anders verkeerd of onvoldoende wordt afgebroken (in werkelijkheid is de lijst langer).

```
\hyphenation{res-pon-se com-man-do ma-na-ge-ment-acti-vi-tei-ten}
```

Omdat ik geen zin heb mijn tekst na te kijken op overfull boxes zet ik `\emergencystretch` op een redelijk grote waarde:

```
\setlength{\emergencystretch}{1em}
```

Een andere actie is het herdefiniëren van de afstand die met `\`, wordt overgeslagen in tekstmode (ik vind deze standaard te groot voor times, bij computer modern voldoet de standaardwaarde beter). De afstand in math-mode laat ik ongewijzigd, omdat formules in times er al enigszins krap uitzien.

```
\def\thinspace{\kern .15em } % was \def\thinspace{\kern .16667em }
\DeclareRobustCommand{\,}{%
  \relax\ifmmode\mskip\thinmuskip\else\thinspace\fi
}
```

In sommige gevallen is het handig een pagina één of meer regels langer of korter te kunnen maken, om zo net een pagina minder te gebruiken of weduwen en wezen (alleenstaande eerste en laatste regels van een alinea) te voorkomen. Met dit doel definieer ik een commando `\erbij`, dat wanneer het op een pagina wordt gebruikt, de lengte van die pagina met het als parameter opgegeven aantal regels verlengt:

```
\newcommand{\erbij}[1]{\enlargethispage{#1\baselineskip}}
```

Af en toe wil ik in de tekst een breuk ($\frac{3}{8}$ of $\frac{123}{597}$) kunnen gebruiken. Het commando `\breuk` zorgt dat deze er enigszins beschaafd uitzien. De opgegeven verplaatsingen zijn sterk lettertype afhankelijk. De hier weergegeven waarden werken redelijk voor een times (de slash zou eventueel nog iets groter kunnen worden gezet).

```
\newcommand{\breuk}[2]{\leavevmode\kern.1em
  \raise.7ex\hbox{\footnotesize #1}\kern-.1em
  /\kern-.05em\lower.25ex\hbox{\footnotesize #2}}
```

Het volgende commando eet zijn argumenten op en doet verder helemaal niets. Dat is op zich ook nog een behoorlijke kunst, spaties om het commando heen moeten namelijk ook genegeerd worden, dat doen de twee hacks.

```
\makeatletter
\newcommand{\opm}[1]{\@bsphack\@esphack}
\makeatother
```

Wat het nut van het hierboven beschreven commando is, is misschien niet iedereen duidelijk. Mijn doel is dat ik in mijn document opmerkingen op kan nemen, die standaard niet worden uitgeprint. Als ik een papieren versie wil waarop mijn opmerkingen wel staan, dan herdefinieer ik `\opm` als volgt:

```
\newcommand{\opm}[1]{\{\@bsphack\par\flushleft\small\em #1\}\@esphack}
```

Oproep

Hierboven heb ik laten zien hoe een relatief groot document er bij mij na verloop van tijd uit gaat zien. Ik heb het syllabusvoorbeeld gebruikt om te tonen welke packages ik handig vind, welke trucs ik gebruik en wat een chaos een document na verloop van tijd wordt. Dit is geen voorbeeld van hoe iets hoort of moet, maar een voorbeeld van hoe iets kan. Het lijkt me interessant om in een volgende MAPS eens aandacht te besteden aan de trucs van anderen. Vele lezers zullen zelf een vergelijkbaar document onder handen hebben dat ook een aparte MAPS-bijdrage verdient. Anderen werken wellicht met minder omvangrijke documenten, maar zullen zich toch ook een aantal trucs eigen hebben gemaakt en een set van favoriete packages hebben. Dit soort zaken en voorbeelden van oplossingen die handiger zijn dan wat ik hier presenteer kan eenieder aan mij of de redactie mailen, zodat we daar in een volgende MAPS één of meer artikelen aan kunnen wijden.

systems

The T_EXLive CDROM easy and fast installation

Wybo Dekker
wybo@servalys.hobby.nl

keywords
T_EX Live, Unix

abstract

Recently all **ntg** members received the latest T_EXLive **cdrom**. This article describes a quick and efficient method of installing the content of this T_EX repository. Since you can download the latest **cdrom** image via the internet, you can also use this method for occasional updates.

If you don't like to make decisions on how and where to install T_EX, you may like to run T_EX straight from the T_EXLive CDROM. If you do so, you have everything there is, but your system will be rather slow.

If you use a Unix operating system, you can use the following script, which provides you with a running system with a minimum of decisions, given that you can afford to create a 680MB file somewhere and have the T_EXLive cdrom in your CDROM drive. You must be root to run this script. On my system (223 MHz, 192Mb) this script took 9 minutes to complete. But,

- you'll probably want to run `texconfig`;
- users who throw `/etc/profile`'s `PATH` setting away will have to take care to include `/texlive/bin/i386-linux` explicitly in their `PATH`;
- the `bs=64M` may be too large if you have not at least that much RAM (use about half your RAM size);
- you may have to replace `/etc/profile` and `i386-linux` with values appropriate for your system;
- as far as I know, similar tricks cannot be done on winxx systems.

The script is as follows:

```
# set your system info:
PROFILE=/etc/profile.local
OS=i386-linux
HALFRAM=64M
# directory with enough space
```

```
SPACYDIR=/space
# copy the cd-image to a file
dd if=/dev/cdrom of=${SPACYDIR}/texlive \
  bs=$HALFRAM
# create a mount point:
mkdir -p /texlive
# add an entry to the file system table:
echo $SPACYDIR/texlive /texlive iso9660 \
  loop=/dev/loop3 0 0 >>/etc/fstab
# mount the new filesystem:
mount /texlive
# create a local tex-tree:
mkdir -p /usr/TeX.local/web2c
chmod a+rw /usr/TeX.local
# set global environment variables:
cat <<EOF>>$PROFILE
export VARTEXMF=/usr/TeX.local
export PATH=/texlive/bin/${OS}:${PATH}
export MANPATH=/texlive/man:${MANPATH}
export TEXMFCNF=/usr/TeX.local/web2c
EOF
# activate those for current session:
. $PROFILE
# correct texmf.cnf:
CNF=$TEXMFCNF/texmf.cnf
cp /texlive/texmf/web2c/texmf.cnf $CNF
grep "% more mem for context" $CNF ||
cat << EOF >> $CNF
% more mem for context and mf:
main_memory.context      = 1500000
hash_extra.context       = 50000
pool_size.context        = 1000000
string_vacancies.context = 90000
max_strings.context       = 100000
pool_free.context        = 47500
nest_size.context        = 500
```

```
param_size.context = 5000 # let tex know where everything is:
save_size.context = 50000 mktexlsr
stack_size.context = 10000 # generate formats for context:
obj_tab_size.context = 200000 texexec --make en nl
main_memory.mf = 800000
EOF
```

pdftex

pdf \TeX 's little secret

Tracking positions

keywords

pdf, pdf \TeX , Con \TeX t, em \TeX , specials, positioning

abstract

It is no secret that pdf \TeX introduces new primitives, but some of them are less known than others. In this article I will describe an example of the use of `\pdfsavepos` cum suis. We will investigate their usage by implementing a simple em \TeX specials simulator.

When Hàn Thé Thành, the author of pdf \TeX , and I were exchanging some emails on pdf \TeX functionality, positional information popped up as potential extension. Actually, it did not take that much time to cook up the basic functionality and the author had implemented it before I could even start to think about real advanced applications.

I'm sure that \TeX programmers can spend many days on how and what kind of information is needed if you want to have access to positions, but since high level macros will probably be used anyway, even things like multiple reference points have proved to be rather unimportant at the system level.

Therefore, pdf \TeX provides just these three primitives:

```
\pdfsavepos  marks the current position
\pdflastxpos the last marked horizontal position
\pdflastypos the last marked vertical position
```

Based on these three primitives, very advanced systems can be build, and for some time now, Con \TeX t has such a system in its core. However, not everyone uses Con \TeX t, so we will demonstrate position tracking in generic applications.

Because pdf \TeX produces its output directly, many of those nice tricks provided by back-ends by means of `\special` fail when producing PDF code directly. Take for instance em \TeX specials. When someone sent me a mail asking if pdf \TeX did support those specials, the original answer was "no", but in the last few years I have learned that you must never underestimate \TeX 's capabilities.

I must admit that I never use those specials myself, but from the way they were used in the macros I was sent, I learned that they depend on the back-end's capability to access the current position. For those who know \TeX this may be bad news, since pure \TeX does not provide any positional information. So in order to use those specials, you must be sure that they are supported by every driver you use. However, the good news is that pdf \TeX does support position tracking, so here is our generic example.

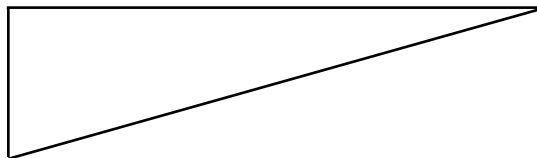
The two em \TeX specials we need to implement are packaged into the macros `\EMmoveto` and `\EMlineto`, like:

```
\def\EMmoveto{\special{em:moveto}}
\def\EMlineto{\special{em:lineto}}
```

They are used in macro packages to draw lines, and the results are often boxes with content like the following:

```
\vbox to 2cm
  {\offinterlineskip \EMmoveto
   \hskip 7cm      \Emlineto
   \vskip 2cm      \Emlineto
   \vskip-2cm      \Emlineto}
```

This box will contain a triangle, and when typeset, it should look like:



These two macros can be implemented as follows. When moving to a position, we only have to register the new coordinates. Once they are known, we use them to draw a line and afterwards we save these end coordinates as starting point for the next line segment. So, at each point specified by `\Emlineto` we need to know the coordinates.

```
\def\EMmoveto%
  {\EMgetposition\Emlastmovex\Emlastmovey}
```

The primitives `\pdflastxpos` and `\pdflastypos` return a number, representing the x and y coordinate in scaled points, TeX's smallest unit of length. We need to convert this number into base points as used by POSTSCRIPT and PDF. When done, we insert some literal PDF code into the text using `\pdfliteral`. Here, the m means 'moveto', the l means 'lineto' and the S operator 'strokes' (draws) the line.

```
\def\Emlineto%
  {\bgroup
   \EMgetposition\Emlastlinex\Emlastliney
   \count0=\Emlastmovex \advance\count0 by -\Emlastlinex
   \count2=\Emlastmovey \advance\count2 by -\Emlastliney
   \divide\count0 by 65536
   \divide\count2 by 65536
   \pdfliteral{1 w 0 0 m \the\count0 \space\the\count2 \space 1 S}%
   \global\let\Emlastmovex\Emlastlinex
   \global\let\Emlastmovey\Emlastliney
   \egroup}
```

We need a fresh start, so we first set the current position to zero.

```
\def\resetEMspecials%
  {\gdef\Emlastmovex{0}\gdef\Emlastmovey{0}}
```

Next comes the macro that keeps track of the position. The current position is marked with `\pdfsavepos` and its coordinates are written to a file whenever the page is shipped out, since `\write` postpones its action until that moment. The file has entries like:

```
\EMsetpos 1 4661756 46651918
\EMsetpos 2 5000359 46990521
\EMsetpos 3 4661756 46313315
\EMsetpos 4 5338962 46990521
\EMsetpos 5 4661756 45974712
```

These lines are written with the command:

```
\write\EMfile
  {\EMsetpos\number\EMcounter
   \space\number\pdflastxpos\space\number\pdflastypos}%
```

In reality the argument to `\write` looks slightly more complicated, because we have to make sure that the number of the current position is frozen and `\EMsetpos` is not expanded. We do so by explicitly expanding the number beforehand and preventing expansion of `\EMsetpos`.

```
\def\EMgetposition#1#2%
  {\bgroup
   \pdfsavepos
   \global\advance\EMcounter by 1
   \expandafter\write\expandafter\EMfile\expandafter
     {\expandafter\noexpand\expandafter\EMsetpos\number\EMcounter
      \space\number\pdflastxpos\space\number\pdflastypos}%
   \EMsetcounters
   \xdef#1{\the\count0}%
   \xdef#2{\the\count2}%
  \egroup}
```

The counter mentioned a few lines ago needs to be declared before it can be used.

```
\newcount\EMcounter
```

We also need a dedicated file slot.

```
\newwrite\EMfile
```

Before we open the file for writing, we read in the data written in the previous pass, but only if the file is present.

```
\def\EMfilename{\jobname.emp}

\def\startEMspecials%
  {\resetEMspecials
   \openin\scratchread=\EMfilename \relax
   \ifeof\scratchread\else \input \EMfilename \relax \fi
   \closein\scratchread
   \immediate\openout\EMfile=\EMfilename\relax}

\def\stopEMspecials%
  {\closeout\EMfile}
```

Just to be sure, we test if `\scratchread` is defined, and if not, we allocate a slot.

```
\ifx\undefined\scratchread \newread\scratchread \fi
```

This leaves us two commands. The `\EMsetpos` command that ends up in the file stores each position in a macro. When this macro is expanded, it assigns the coordinates to two scratch counters.

```
\def\EMsetpos#1 #2 #3 % number x y
  {\expandafter\xdef\csname EM:#1\endcsname{\count0=#2 \count2=#3}}
```

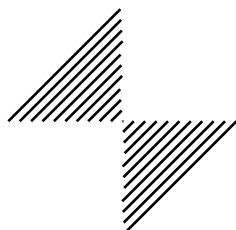
This position is recalled with its companion macro. First we set the counters to zero. When the position is unknown, nothing happens since the `\csname...` will expand to `\relax`.

```
\def\EMsetcounters%
  {\count0=0 \count2=0
   \csname EM:\the\EMcounter\endcsname}
```


These macros are rather independent of the macro package you use. For instance, in `CONTEXT` the following works well:

```
\setuppositioning[unit=ex]
\startpositioning
  \dostepwiserecurse{-10}{10}{1}
  { \position(0,\recurselevel) {\EMmoveto}
    \position(\recurselevel,0) {\EMlineto} }
\stoppositioning
```

Here, we hook the `EMTeX` macros into an existing text positioning mechanism, which positions the commands using `TeX`'s skips and kerns.



Of course one should start and end the file with:

```
\startEMspecials
\stopEMspecials
```

and, if needed, reset the begin position at each page using:

```
\resetEMspecials
```

A few pages ago, we mentioned that `CONTEXT` has built-in position tracking. This means that when we want to implement this kind of trickery in this macro package, we can fall back on existing functionality. In the following alternative we will also use a few skips. This keeps the source readable and `CONTEXT` has plenty of unused registers to accommodate this strategy.

```
\newcount\EMcounter \def\EMvariable{EM:\the\EMcounter}
\newskip \EMlastmovex \newskip \EMlastmovey
\newskip \EMlastlinex \newskip \EMlastliney
\def\resetEMspecials%
  {\global\EMlastmovex=0pt \global\EMlastmovey=\EMlastmovex}
\resetEMspecials \appendtoks\resetEMspecials\to\everyshipout
```

Watch how we reset the specials after a page is flushed. We don't have to bother about files here, because saving and recalling is already implemented. Although not needed, we define the start-stop macros, so that `CONTEXT` users who key them in are not confronted with error messages.

```
\let\startEMspecials\relax
\let\stopEMspecials \relax
\def\EMgetposition#1#2%
  {\global\advance\EMcounter 1
   \setposition\EMvariable
   \global#1=\POSx\EMvariable
   \global#2=\POSy\EMvariable}
```

```

\def\EMmoveto%
  {\EMgetposition\EMlastmovex\EMlastmovey}

\def\EMlineto%
  {\EMgetposition\EMlastlinex\EMlastliney
  \global\advance\EMlastmovex -\EMlastlinex
  \global\advance\EMlastmovey -\EMlastliney
  \ScaledPointsToBigPoints{\number\EMlastmovex}\EMx
  \ScaledPointsToBigPoints{\number\EMlastmovey}\EMy
  \pdfliteral{1 w 0 0 m \EMx \space \EMy \space l S}%
  \global\EMlastmovex\EMlastlinex
  \global\EMlastmovey\EMlastliney}

```

The command `\setposition` registers a position by name (here `\EMvariable`), while `\POSx` and `\POSy` give you access to the coordinates.

These three commands are contained in a suite of low level commands that can be used to register and get access to positional information. The current mechanism is not yet complete, but already provides enough hooks for advanced embedded graphics. Its functionality is a natural extension to the METAPOST support already present in `CONTEXT`. Therefore, more advanced examples can be found in the METAFUN manual, since they fall beyond the scope of this introductory article.

As a bonus, I will now provide a few macros that will make this mechanism transparent to DVI as well as PDF output. We will use `\pdfoutput` as trigger.

```
\ifx\pdfoutput\undefined \chardef\pdfoutput=0 \fi
```

We save some of the macros we defined previously:

```

\let\pdfEMmoveto\EMmoveto
\let\pdfEMlineto\EMlineto

\let\pdfstartEMspecials\startEMspecials
\let\pdfstopEMspecials \stopEMspecials

```

We now redefine them to support DVI and PDF.

```

\def\EMmoveto%
  {\ifcase\pdfoutput\special{em:moveto}\else\pdfEMmoveto\fi}
\def\EMlineto%
  {\ifcase\pdfoutput\special{em:lineto}\else\pdfEMlineto\fi}

\def\startEMspecials%
  {\ifcase\pdfoutput\else\pdfstartEMspecials\fi}
\def\stopEMspecials%
  {\ifcase\pdfoutput\else\pdfstopEMspecials \fi}

```

If there is any real demand for this in `CONTEXT`, I will hook these macros in the special drivers, so that their support becomes more natural.

You may wonder to what extent positional tracking is `PDFTEX` specific. In `CONTEXT`, we also support position tracking in DVI by using specials and analyzing the DVI file afterwards using a PERL script created by Taco Hoekwater. Since many of the advanced `TEX` features depend on some kind of back-end, we don't consider it to be a disadvantage. Of course, the `PDFTEX` way is not only cleaner, but also faster. It was more out of curiosity than out of need that we provided the DVI methods as well. Also, it is always good to have more roads to reach the same goal.

keywords

pdf, fdf, annotations, pdfTeX, ConTeXt

abstract

This article describes how to use pdfTeX to typeset annotations added to pdf documents. This method uses ConTeXt but can be applied to any valid pdf document.

The ACROBAT viewers support text annotations. These are small notes that popup when the user opens them. There are several ways to add annotations to a document. For example, in ConTeXt you can say:

```
\startcomment
  Hello beautiful\world!
\stopcomment
```

Annotations use the PDF document encoding. Fortunately, we can map most common accented characters onto this encoding vector. Therefore, the next sequence of characters will show up properly in viewers. As you can see, annotations can have certain properties, like titles (first argument), size and color.

```
\startcomment[accents][color=green,width=4cm,height=3cm]
  We love \'a\cc\cc\'e\~nt\SS
\stopcomment
```

In ConTeXt, annotations are automatically placed in the margin. Multiple annotations in the same line will be stacked in such a way that they are all visible.

I must admit that until now I seldom used this mechanism, if only because annotations are not supported by GHOSTVIEW. On the other hand, one of my enthusiastic and dedicated proofreaders, David Arnold, always sends his comments as annotations.

Now, I may like to design documents for displays, but reading them from today's screens is not always a pleasure. Because I wanted to print the annotations, I decided to write a module to do that for me. We have been using this module for some time now, and this method of annotating text and printing that part of the original that is annotated along with its annotations works well. At this moment, the functionality is limited, but new options may be added in the future.

Given that you have a file with annotations, here is what we need to do:

- Export the annotations to a file with the same name as the PDF file, but use the suffix fdf instead. You may consider FDF to be the database format related to PDF.
- Run the PERL script fdf2tan with the filename as argument (no suffix).
- Process the resulting file fdf-tan with ConTeXt (TeXEXEC).
- You can now print (or view) the result in fdf-tan.pdf.

The resulting file has two parts. The first part contains pages of the original that have annotations. The annotated locations are marked with a number. The second part shows the text of the annotations. You can click on the number of the annotation and go to the place where it was attached. The numbers in the first part and those in the second part are cross linked.

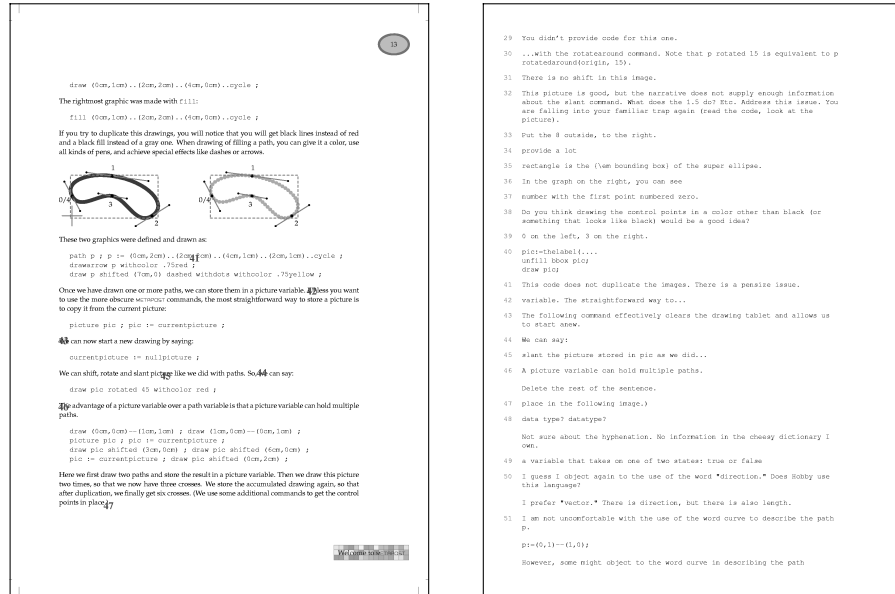


Figure 1 The copied page and the annotations are typeset in one document and cross-linked to each other.

I already mentioned that encoding is an issue. Both the PERL script and the T_EX macros try to make the best of whatever they may find in the annotation. Although I do have an FDF parser written in PERL, I prefer to use PERL for this purpose, if only because it is more tolerant to the unexpected. Of course we use PDF_TE_X to typeset it all nicely. You can process any PDF document, but you must have CON_TE_XT installed on your system in order to typeset the file fdf-ann.tex.

Hanging punctuation

a pdfTeX microtypographic extension

Hans Hagen
Ridderstraat 27, 8061GH
Hasselt
pragma@wxs.nl

keywords

micro typography, margins, kerning, pdfTeX, ConTeXt

abstract

At the time of this writing, Hàn Thế Thành is writing his thesis, wherein he explains in great detail two extensions to TeX's typesetting engine. In this article we will explore marginal kerning, or character protruding. We will demonstrate the effects and introduce the new primitives, as well as explain how to enable this feature in ConTeXt.

Introduction

When breaking a paragraph into lines, only a few typographic engines consider the paragraph as unit of information and TeX is one of them. Although TeX has a pretty good reputation when it comes to breaking paragraphs, it is no secret that micro-typographic improvements are not easily achieved.

After some years of experimenting, as part of his promotion, Hàn Thế Thành, the author of pdfTeX, has extended TeX's core line breaking algorithm with two micro-typographic features: *hz* optimization and marginal kerning. As soon as you know that the acronym *hz* represents the name of a typographer, the name of Hermann Zapf will spring into mind. In the spirit of Gutenberg, he has invented a method for visually improving the grayness of a paragraph by slightly changing the shape of a glyph.

A maybe less impressive, but certainly worthwhile second extension to the TeX typesetting engine is marginal kerning, also called protruding. Where *hz* optimization demands additional font resources, and thereby is more complicated to support, protruding can be supported with minimal effort from the user. Therefore, in this article, we will focus on protruding and leave *hz* for another article, if only because it would put some demands on the production process of the MAPS.

Protruding characters

In the following fake paragraph, you can see a hyphenation point, a secondary sentence, separated by a comma, and a last sentence, ending with a period. Miraculously, this paragraph fits into lines. Although exaggerated, these lines demonstrate that visually the hyphen and punctuation characters make the margin look ragged.



Before computers started to take over the traditional typesetter, it was common practice to move hyphens and punctuation into the margin, like in:





In this alternative, the overall appearance is more gray, and therefore enhances the process of reading.

Sometimes, shifting the characters completely into the margin is too much for the sensitive eye, for instance with a slanted font, where the characters already hang to the right. In such cases, we need to compromise.



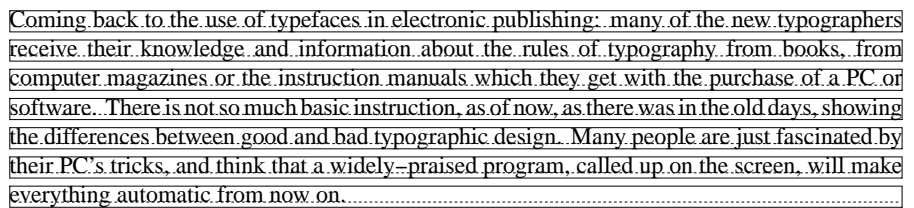
In PDF_T_E_X protruding is implemented in two ways. The first method does not take protruding into account when breaking the paragraph, but the second method does. In method one, the characters are moved into the margins after the paragraph is broken, and this move influences the inter-word spacing. As said, the second method takes the shifts into account when breaking the paragraph into lines. This means that when method two is enabled, the line-breaks may differ from the normal situation.

Since we want optimal typesetting, we will focus on protruding method two. For this purpose we will use a quote from Hermann Zapf's article "About micro-typography and the *hz*-program" in *Electronic Publishing*, vol 6 (3), 1993.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

After T_EX has typeset this paragraph, it has constructed the following lines.



As you can see, the height and depth of the lines depend on the characters, but their width equals what T_EX calls `\hspace`. However, the natural width of the lines may differ from `\hspace`.

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

Here the inter-word space is fixed to what T_EX considers to be a space. This example also demonstrates that T_EX does not have spaces, but stretches the white area between words to suit its demands. When breaking lines, T_EX's mind is occupied by boxes, glue and penalties, or in more common language: (parts of) words, stretchable white space, and more or less preferred breakpoints.

<p>Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the in-</p>	<p>struction manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design.</p>	<p>Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.</p>
--	--	--

This time we have enabled PDF_TE_X's protruding mechanism. The characters that stick into the margin are taken into account when breaking the paragraph into lines, but in the final result, they do not count in the width. Here we used an ugly three column layout so that we got a few more hyphens to illustrate the principle, but in the next examples we will stick to two columns.

<p>Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic</p>	<p>instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.</p>
---	---

In this first example we just typeset the text in the traditional way. The hyphens and punctuation fit into the margin.

<p>Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic</p>	<p>instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.</p>
---	---

In this example, the protruding machinery is put to labour. The hyphens and punctuation may now stick into the margin completely. The next two examples shows what happens when we limit the protruding to 75% and 50% respectively.

<p>Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction</p>	<p>manuals which they get with the purchase of a PC or software. There is not so much basic instruction, as of now, as there was in the old days, showing the differences between good</p>
--	--

and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on

Coming back to the use of typefaces in electronic publishing: many of the new typographers receive their knowledge and information about the rules of typography from books, from computer magazines or the instruction manuals which they get with the purchase of a PC or software. There is not so much basic

the screen, will make everything automatic from now on.

instruction, as of now, as there was in the old days, showing the differences between good and bad typographic design. Many people are just fascinated by their PC's tricks, and think that a widely-praised program, called up on the screen, will make everything automatic from now on.

Although available in PDF \TeX , this feature is not limited to PDF output. Also, when protruding is not enabled, or when level one protruding is used, the output is 100% identical.

The new primitives

The way protruding is implemented is, as you may expect from a \TeX extension, rather advanced. Although such features make most sense when encapsulated in the font switching mechanism, we will now introduce the new primitives. First we define a font, say a Times Roman, using \TeX 's `\font` primitive.

```
\font\NiceFont=tir at 12pt
```

For this font, we want to stick the hyphen and some punctuation into the right margin. The `\rptcode` primitive (globally) assigns a factor to a specific character slot in a font. The `\-` is \TeX 's way of calling up the character number associated with the hyphen.

```
\rptcode\NiceFont\ -=700
\rptcode\NiceFont\ .=700
\rptcode\NiceFont\ \ ,=700
\rptcode\NiceFont\ \ :=700
\rptcode\NiceFont\ \ ;=700
```

A factor of 700 means that the hyphen will move 70% of its width into the (here) right margin. You can specify the left protruding factor in a similar way using the `\lptcode` primitive.

These definitions demonstrate that each character can be set up to protrude, so we can say:

```
\rptcode\NiceFont\ \ !=200
```

This will move the exclamation mark some 20% into the right margin when it ends up there. In this way, for each font dedicated shifts can be specified. Although hanging punctuation can be faked with macros to a certain extent, it will be clear that the fact that you have control over each character, combined with tight integration in the paragraph builder, beats every macro based solution.

So far, we have only defined protruding factors. The protruding mechanism itself is enabled with the low level primitive `\pdfprotrudechars`. When set to 1, you get level one protruding, which does not influence the line breaks, but for optimal results, you should enable level two protruding.

Support in Con \TeX T

At this moment, Con \TeX T supports protruding as an experimental extension to its font switching mechanism. Since protruding is related to the shape of the font, setting it up (currently) takes place in a similar way as encodings.


```
\definefontsynonym [Times-Roman] [tir] [encoding=ec,handling=normal]
```

Of course it would be best to make protruding specification part of the font specification. You can read more about this in Hàn Thế Thành's thesis, where he also reports the opinions of Donald Knuth and Hermann Zapf on his extensions as well as some alternative approaches.

Defining a protruding vector in `CONTEX`T does not differ much from other encoding specific definitions. The default punctuation vector looks like:

```
\startfonthandling[defpunc]
  \defineprotrudefactor ! 0 .2
  \defineprotrudefactor , 0 .7
  \defineprotrudefactor - 0 .7
  \defineprotrudefactor . 0 .7
  \defineprotrudefactor : 0 .5
  \defineprotrudefactor ; 0 .5
  .....
\stopfonthandling
```

Instead of using numbers (like the 700 in the previous definition), we use fractions, one for the left and one for the right shift. There are a few more vectors defined, like `defalph` for characters, where we only apply very small shifts.

A combination of such vectors is packaged in a font handler using the following command:

```
\definefonthandling [normal] [defpunc,defalph]
```

Since these definitions depend on the encoding, they are loaded with the `\useencoding` command, for instance:

```
\useencoding[pro]
```

Again, in `CONTEX`T we don't use the low level primitive `\pdfprotrudechars` directly, but we have integrated it into the normal alignment macros. Apart from other advantages, this means that we can guarantee compatibility with other `TEX`'s.

```
\setupalign[hanging] % in dutch: \steluitlijnenin[hangend]
```

Font protruding is turned off automatically in controlled situations, and more control will be added in due time. We already mentioned that the amount of protruding depends on the shape, which is why we may need different values for slanted and bold shapes. One way of using the same vector for different shapes, is changing the strength of the protruding:

```
\definefonthandling [slanted] [defpunc] [right=1.5]
```

Here, because a slanted glyph already sticks into the margin, we limit protruding to punctuation.

It will take some time and experiments to define suitable vectors for different classes of shapes and determine the optimal values for the strength. But, if this feature is found useful, you may count on `CONTEX`T to come with a suitable set of such vectors. In spite of what you may expect from the previous examples, the user can limit his or her effort to enabling this nice feature by `\setupalign`.

Bogusław Jackowski
BOP s.c.
ul. Piastowska 70
80-363 Gdańsk, Poland
B.Jackowski@gust.org.pl

Janusz M. Nowacki
FOTO-ALFA
al. 23 Stycznia 56d
86-300 Grudziadz, Poland
J.Nowacki@gust.org.pl

Piotr Strzelczyk
BOP s.c.
ul. Piastowska 70
80-363 Gdańsk, Poland
piotrs@telbank.pl

fonts

Antykwa Półtawskiego: a parameterized outline font

editor's note

This paper originally appeared as 'Antykwa Półtawskiego: a parameterized outline font', pages 117–141, in *EuroTeX'99 Proceedings Paperless T_EX*, Günter Partosch and Gerhard Wilhelms (eds.), s.l., 1999

No doubt, METAFONT is a powerful programming language, well-suited for designing fonts, in many respects much better than popular WYSIWYG programs (precision, possibility of complex constructions, etc.); and, no doubt, there are thousands of fonts used all over the world, only a negligible fraction of them being designed using METAFONT.

Computer Modern inheritance

The final exhortation of *The METAFONT book* ([1]): "GO FORTH now and create *masterpieces of digital typography!*" suggests that Donald E. Knuth, when he designed METAFONT, presumed that his idea of parameterized fonts would find many followers. Unfortunately, his expectations failed. Why? The reasons are manifold.

One of the most important aspects is perhaps the exceptional programming talent of Knuth – his style is not so easy to follow. The family of Computer Modern fonts is very intricate: more than hundred files containing nearly a megabyte of METAFONT code. *They are rather complex* – Knuth modestly admits in *The METAFONT book*.

The question arises whether the complexity of the Computer Modern family reflects the nature of the problem (i.e., type design) or rather Knuth's personal traits. We would incline to the latter opinion. A herd of 62 parameters may raise doubts, the more so as they control not only dimensions and slanting, but even the presence of serifs. Questionable also is Knuth's design decision to keep the continuous change of the proportions of glyphs along with the change of font size – it perceptibly deteriorates the quality of glyphs in smaller fonts (5–7pt).

Knuth was apparently aware of weak points of the Computer Modern design. In *The METAFONT book* he admits: *they [the Computer Modern typefaces] were developed rather hastily by the author of the system, who is a rank amateur at such things.*

Parameterization was certainly a great idea, but it seems that Knuth went too far in exploiting it.

Bitmaps versus outlines

Most harmful to the potential success of METAFONT as a tool for type designers was perhaps Knuth's adherence to the bitmap representation of fonts. Although in METAFONT all graphic objects are represented by outlines (Bézier curves), METAFONT's primary task is to generate bitmaps. Therefore the outline representation of glyphs was unimportant for Knuth.

For example, the letter 'S' of *cmr10* consists of five pieces filled and, worse still, stroked with a circular pen (Fig. 1). Many tricks of this kind can be found among the Computer Modern programs: stroking with various pens, erasing (dish serifs), copying bitmaps (German double 'S' in *cmesc10*), etc.

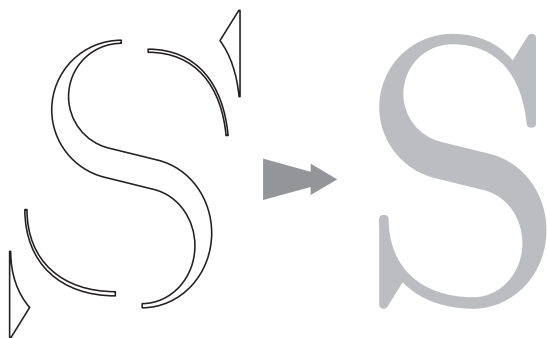


Figure 1. The construction of the letter ‘S’ of *cmr10*: five separate elements are stroked with a pen and filled.

Such an approach is satisfactory as long as the resulting bitmap is the main concern. The fact is that $\text{T}_{\text{E}}\text{X}$ bitmap fonts have not become a worldwide standard. Instead, outline font formats, *Type 1* (PostScript) and *TrueType* (Windows; its PostScript equivalent is known as *Type 42*), have won the competition.

The problem is that there does not exist a simple method of conversion of METAFONT bitmap-oriented programs into a purely outlined form. Some preliminary results (by Andreĭ Slepukhin, [2]) show that it could be devised specifically for the Computer Modern family, but:

1. the reasonableness of such an effort is doubtful;
2. the converter can be neither efficient nor reliable.

Richard J. Kinch, a staunch devotee of outlines ([4], p. 134) developed an interesting tool for an interactive conversion of Computer Modern fonts to an outline form, *MetaFog* ([3]). Recently, several fonts were prepared by Taco Hoekwater ([5]) using this technology.

In spite of the successes of Kinch’s approach, it looks as if re-writing the Computer Modern programs from scratch were more advisable. Still better would be to have a macro package facilitating the *creation* of outline fonts. But is METAFONT the most adequate tool for such a purpose?

METAFONT versus METAPOST

In 1989, five years after the first release of METAFONT¹, METAPOST ([6]) came to this world. The originator was John D. Hobby, who designed many of the elegant algorithms employed in METAFONT. Hobby realised that METAFONT is an excellent tool for designing graphics, not only fonts, and that bitmap output is a severe limitation. His idea was to use the METAFONT language to create PostScript output. He did not consider, however, making a tool for generating PostScript fonts. Fortunately, his adaptation was sufficiently general to admit font applications as well.

Again, a question arises: does it make sense to force METAPOST to do things for which it was never intended? The answer is equivocal.

There are some interesting features present in METAFONT and absent from METAPOST, and vice versa. For example, the measuring of arc length is absent from METAFONT.

1. Actually, the first version of METAFONT appeared in 1979. Having gathered experience, Knuth released a new version of METAFONT in 1984, re-written from scratch and incompatible with the predecessor. In the source of METAFONT, *mf.web*, the history of METAFONT starts with the statement: *Version 0 was completed on July 28, 1984.*

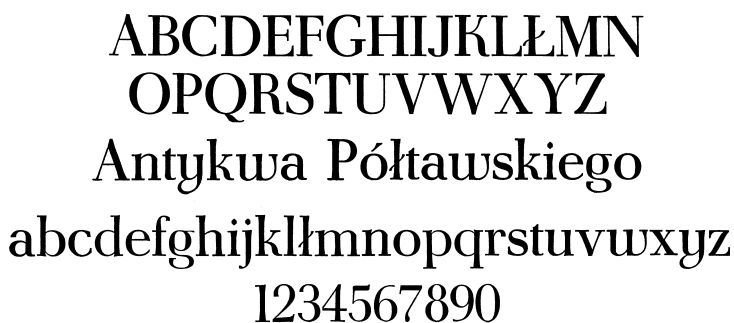


Figure 2. Antykwa Półtawskiego: letters and digits (lead print).

FONT and present in METAPOST, whereas METAFONT, but not METAPOST, is capable of measuring the area tinted by filling a cyclic path.

From the point of view of the generation of outline fonts, both programs need postprocessing: with METAFONT one has to analyse either a generic font file or a log file; with METAPOST the resulting EPS files are to be processed.

It is intuition that remains in such ambiguous situations – it told us: METAPOST.

Antykwa Półtawskiego

The best method of the verification of intuition is to deal with a real-life case. None of the authors is a professional type designer, so we could not even dream about designing a font *ab ovo*. Fortunately, there exists an elegant typeface, designed in the 'twenties by a Polish typographer, Adam Półtawski (Fig. 2). It was a tempting challenge to test our intuition against Półtawski's typeface.

In the sequel we shall use the Polish name "Antykwa Półtawskiego" instead of the English translation "Półtawski's Antique," as the word "antique" – according to the Collins dictionary – means: *a family of typefaces having all lines of nearly equal thickness*. This is not true with Półtawski's design. As you can see, it resembles rather neoclassical (also called modern) typefaces.

At first glance, Antykwa Półtawskiego looks very regularly, almost austere. A close inspection, however, reveals many surprising details (Fig. 3).

Studying the imprints of Antykwa Półtawskiego, we found ourselves in the position of the explorers of pyramids: we had to conjecture the rules governing shapes and propor-



Figure 3. Antykwa Półtawskiego: characteristic details and features. Observe polygonal dots, unconventional letters 'g', 'y' and 'w,' and queer right-angled hooks in 'j' and 'f'. Note also the neatly stroked ribbon-shaped diacritical elements of the Polish 'l' and 'L' – certainly, they cannot be called "slashes."

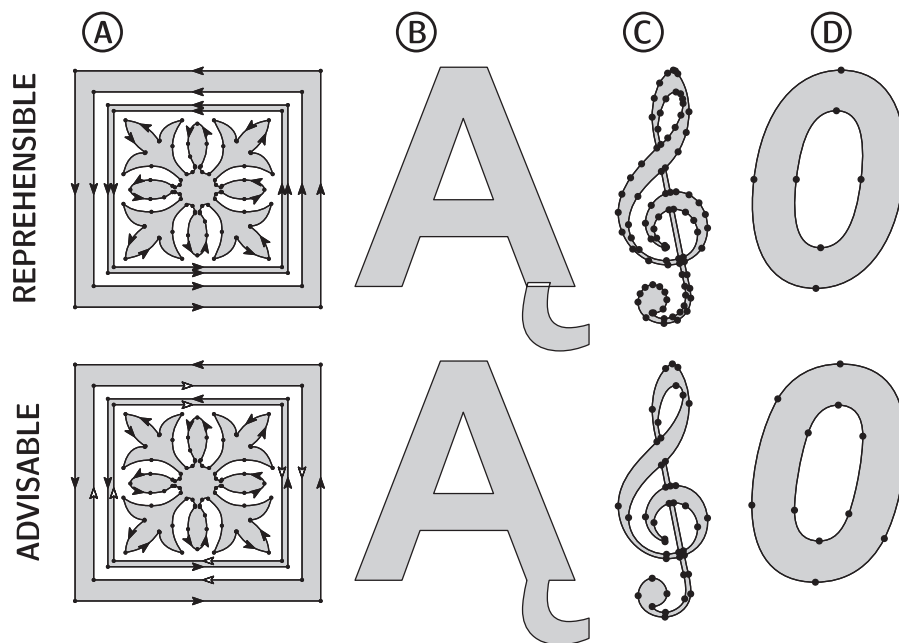


Figure 4. Common faults of outline design and avoiding them; the letter ‘Aogonek’ in the top line comes from the standard set of Windows fonts (*Arial CE Bold*).

tions from faint evidence (or even none). Our situation would improve if we had access to the original Półtawski templates. So far, we have not managed to track them down, but there remains a spark of hope that they exist somewhere.

The bright side of this otherwise confusing situation is that it compels us to looking for a general and flexible approach. Otherwise, we could easily get lost among the peculiarities of Antykwa Półtawskiego.

What we actually expect to obtain is a family of typefaces (not as broad as Computer Modern) resembling Antykwa Półtawskiego. If we have luck, one member of the family may happen to replicate nicely the original Antykwa Półtawskiego design, but it is not crucial. Much more important is the possibility of generating font variations on the fly.

A glance at outline paradigms

A compendious survey of technical aspects concerning outline fonts can be found in the Adobe manual concerning *Type 1* fonts ([7]); we summarize here only the most important and general (PostScript-independent) ones.

Direction of paths. In *Type 1* fonts, paths to be filled and to be unfilled have different orientation: the former – anticlockwise, the latter – clockwise (Fig. 4A). We decided to follow this convention, although neither METAFONT nor METAPOST requires it.

Overlapping paths. Overlapping paths should be banished forever from outline fonts. It should be noted, however, that many font vendors, including the celebrated Microsoft, distribute fonts that do not follow this fundamental rule (Fig. 4B), which may cause unwanted effects. Therefore the operation of joining two outlines should belong to the set of basic tools for generating outline fonts (cf. Sec. *Removing overlaps* below).

Conciseness. The outline of a glyph should contain as few segments as possible (Fig. 4C). For example, consecutive co-linear segments should be replaced by a single one. Users of METAFONT and METAPOST should be aware of the problem of “tiny segments”: joining two paths obtained by an intersection operation (`intersectionpoint` or `interseciontimes`) may lead to nearly coincident and thus superfluous nodes. It is an admittedly subtle problem, yet important from the point of view of the construction of tidy outlines (cf. Sec. *Joining paths* below).

Points at extremes. The number of segments should not be too small either. Putting nodes at extremes helps rasterizers to transform glyphs accurately. In particular, repositioning nodes in a slanted font may prove beneficial (Fig. 4D). This implies that a single segment of a Bézier curve should not turn by more than 90 degrees. The points of inflection should also be avoided.

A glance at tools for assembling outlines

The set of tools for generating outlines should facilitate complying with the rules summarized in the previous section. We describe here a few representative tools we implemented. Hopefully, it should suffice to convey the most important issues.

Joining paths. The simplest tool is perhaps an operator for joining paths that have ends nearly coincident:

```
def && = amp_amp_ whatever enddef; % a common postfix-notation trick
%
tertiarydef p amp_amp_ q = % |length(p)>0|
  (subpath(0,length(p)-1) of p) ..
  controls (postcontrol length(p)-1 of p)
  and (precontrol length(p) of p) ..
enddef;
```

You use this operator like a normal ampersand, e.g.:

```
p && q && cycle
```

The only difference is that if the edge nodes do not *exactly* coincide, you must not use a single ampersand; double ampersand works in both cases: it simply removes the former of two nodes. (Note that if nodes are actually distant, you may obtain weird results.)

In the METAFONT sources of Computer Modern, tiny segments appear regularly whenever two paths intersect, for example in the arrows of *cmxy10* (Fig. 5). Such segments are harmless from the point of view of generating bitmaps. They can, however, cause quite a lot of commotion from the point of view of outline construction: the path shown in Fig. 5 yields strange values of the `turningnumber` function, depending on the actual resolution: -1 for a laser printer mode (300 dpi), and -3 for a phototypesetter mode (1270 dpi).

It is numerical instability that is responsible for the singular behaviour of METAFONT and METAPOST. Tiny segments may form loops having an impact on the result of the `turningnumber` operation; thus, the important characteristic of a path, its turning number, is not reliable if paths are constructed incautiously.

Tangency is another feature of geometric objects that leads to numerical instability. For example, the turning number for a path defined as

```
(0,0)--(0,-1){up}..(1,1)--cycle
```

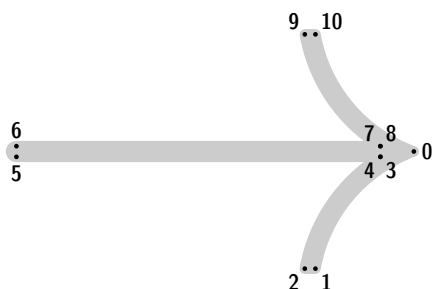


Figure 5. The rightward arrow from *cmsy10*. The outline of the arrow contains two pairs of nearly coinciding nodes: 3, 4 and 7, 8. Their coordinates for a Linotronic mode, 1270 dpi, read: (150.4547, 42.00003), (150.4547, 41.99976), (150.4547, 46.00024), and (150.4547, 45.99997), respectively, which causes a strange turning number of the path (equal to -3).

is 0, whereas the value becomes 1 after reversing the path. The reason behind this somewhat-amazing result is that the path turns by 180 degrees at the node $(0, -1)$, i.e., the path is locally self-tangent.

In general, the problems caused by numerical instability cannot be solved automatically: only a careful control over the details of the construction of a path may help to overcome the unwanted consequences of the instability. This is exactly the reason why a universal, efficient and reliable converter from a bitmap-oriented METAFONT program into a tidy outline form is hardly imaginable (cf. Sec. *Bitmaps versus outlines* above).

A problem in elementary geometry. Elementary geometry prompts constructions that prove useful in computer type design. An example of such a construction is the computation of a side of a right-angled triangle, given its hypotenuse (c) and the length of one of its sides (b):

```
tertiarydef c side b = % |pair c; numeric b;|
  begingroup
    save a; pair a;
    % |(length(c)+-+b)=length(a)|
    c=a+b/(length(c)+-+b)*(a rotated -90);
    a
  endgroup
enddef;
```

The following example illustrates the application of the macro `side`: given the height h and width w of a rectangle, construct a band (parallelogram) of the width b inscribed in the rectangle (Fig. 6A).

The construction with a ruler-and-compasses method is elementary (Fig. 6B):

1. draw the rectangle in question (let's call its diagonal c);
2. draw in one of the corners of the rectangle (say lower-right – node 0 in Fig 6C) a circle of radius b ;
3. draw the side a , i.e., a straight line tangential to the circle and passing through the opposite end of the diagonal (node 2).

The crossing point of the side and the base of the rectangle is one of the two remaining corners of the parallelogram (node 3); the other remaining corner (node 1) can be found similarly.

This construction can be translated to METAPOST as follows:

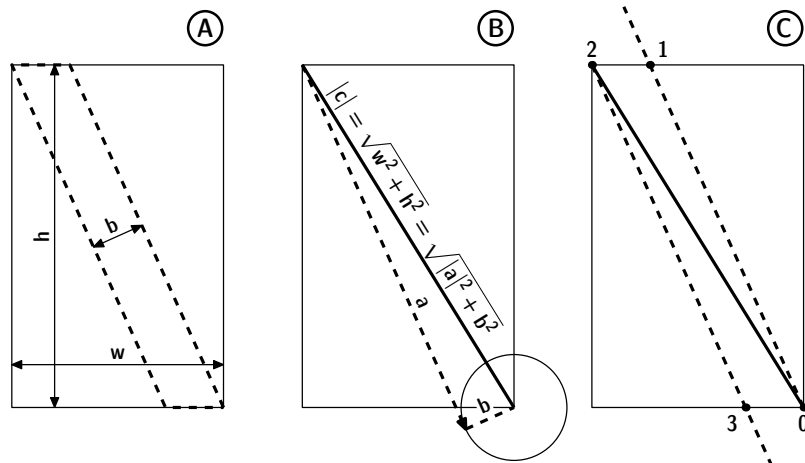


Figure 6. Inscribing a band (a parallelogram) in a given rectangle: a typical example of construction of a side of a right-angled triangle; more explanations in the text.

```

z0=(w,0); z2=(0,h);
z1=z0+whatever*(z2-z0 side -b); y1=y2; % this fixes node 1
z1-z2=z0-z3; % this fixes node 3 by symmetry
    
```

The clue is the expression $z_0 + \text{whatever} * (z_2 - z_0 \text{ side } -b)$ which can be interpreted as “somewhere on a straight line covering the side a .”

Slanting: another problem in elementary geometry. Slanting (in the art of type design) is not as simple an operation as it may look at first glance. We already noted that slanting necessitates adding points at extremes (Fig. 4D). Furthermore, slanting should not affect the width of stems (see Fig. 7A).

Assume that the stem slope is given by a vector d and that its resulting breadth after

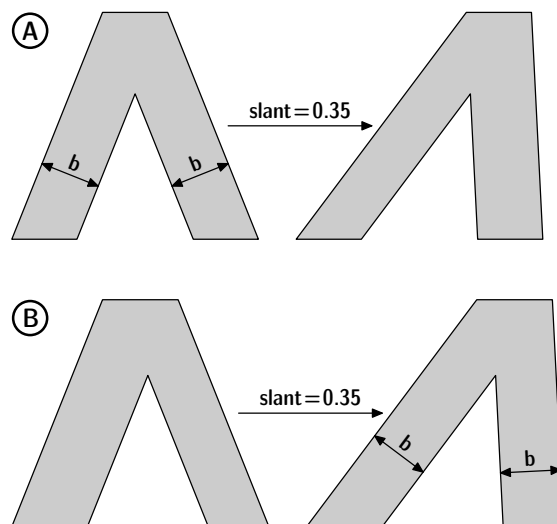


Figure 7. A capital lambda from an imaginary grotesque font: slanting (exaggerated here) usually affects the width of stems (A), but this effect can (and should) be neutralized (B).

slanting by a slant s should be b ; the initial breadth of the stem (i.e., prior to slanting), b' , is given by the following formula, in the sequel referred to as the “slant correction formula”:

$$b' = b * \text{length}(\text{unitvector}(d) \text{ slanted } s)$$

The result of taking this correction into account is displayed in Fig. 7B.

An advanced problem in elementary geometry. Consider now a “mixture” of two previously demonstrated problems: assume that we want to inscribe a band into a given rectangle in such a way that after slanting by a given slant it will have the width b . Neither the macro ‘side’ nor the formula for the correction of a stem width can be applied explicitly, as both initial and final slopes depend on each other.

The solution is an iterative algorithm:

1. use the macro ‘side’ to find the band of the width b for the non-slanted case;
2. find the corrected stem width b' using the slant correction formula;
3. set $b \leftarrow b'$ and repeat the steps (1)–(3) until the process converges (in practice 3–4 steps suffice).

The code is here, in case you’re curious:

```
primarydef c /\ b =
% A variant of the |side| procedure that iteratively counteracts
% a slant deformation; given: |slant|: a slant (global),
% |c|: the hypotenuse (vector) of a right-angled triangle,
% |b|: the length of one of its sides; result: the other side
% of the triangle (vector).
if slant=0: (c side b) else:
begin group save b_,b__, n; b_:=b:=b; n:=10;
  forever:
    b_:=b*length(unitvector(c side b_) slanted slant);
    exitif (abs(b_-b__)<.01) or (n<=0);
    b__:=b_; n:=n-1;
  endfor
  if (abs(b_-b__)>=.01): errmessage "Iteration hasn't converged"; fi
  c side b_
end group
fi
enddef;
```

Examples of the use of this macro can be found in many places in the sources of Antykwa Półtawskiego.

Removing overlaps. As the last general-purpose tool, we consider the operation of “welding” outlines, also known as “removing overlaps.” Most WYSIWYG graphic programs are equipped with such an option. Surprisingly enough, this operation is missing from the set of standard tools for METAFONT/METAPOST.

The set of macros implementing the relevant outline operations, ROEX ([8], [9], [10]), has never become popular (we know about a few people using it) which is understandable, considering the size of the code (70 kB) and the relatively low efficiency and robustness of the employed algorithms.

It should be emphasized, however, that algorithms accomplishing operations of this kind cannot be fully reliable because of the above-mentioned numerical instability problems (cf. Sec. *Joining paths* above). Nonetheless, when used in a disciplined way, they

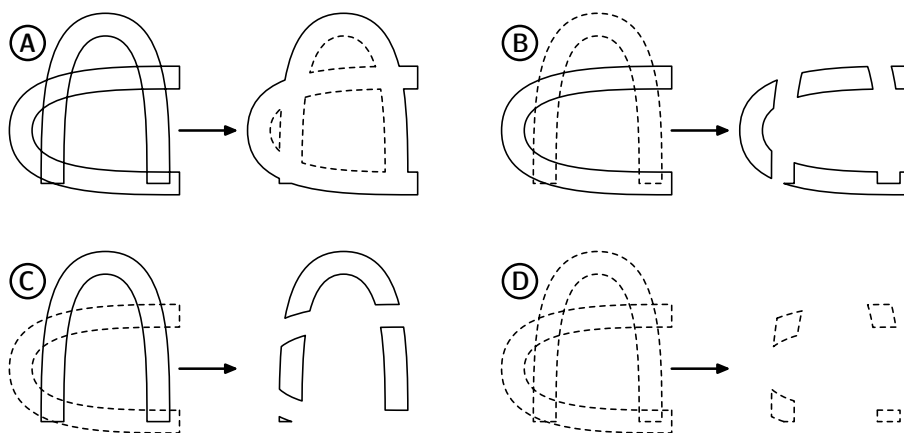


Figure 8. Rudimentary set-theoretic operations on outlines: union (A), subtraction (C, D), and intersection (D); the operations are accomplished by a single macro, depending on the orientation of paths: anticlockwise paths are marked with a solid line, clockwise with a dashed line.

prove to be extraordinarily useful: the source code is compact and comprehensible. Therefore, for the purposes of work on *Antykwa Półtawskiego*, we devised a simplified version (4 kB) of the algorithm accomplishing set-theoretic operations (Fig. 8).

The relevant macro takes two cyclic paths as arguments, intersects them and assembles the result according to the orientation of the input paths: if both paths are anticlockwise, the result is the union of the two paths, if both are clockwise – the result is the intersection, otherwise the clockwise path is subtracted from the anticlockwise one. In fonts, the first situation occurs most frequently (‘x’ or ‘lslash’), but sometimes, due to the paradigm of having two input paths only, other operations can also be necessary (‘oslash’).

Specific tools for *Antykwa Półtawskiego*

So far, we have demonstrated a set of basic tools expediting the work on type shapes. Obviously, each font has its own peculiar yet regular features, having an impact on the font appearance as a whole. The glyph programs should reflect all regularities of the font.

A “building blocks” paradigm. Our general aim was to create a set of “building blocks” – macros returning paths rather than a set of relations (dependencies) such as, e.g., the plain macro `penpos` returns – and a set of operators for handling the paths. The glyphs are assembled from the “building blocks” with the help of the operators.

Let’s look into the code of the letter ‘H’ (the numbering of lines, of course, does not belong to the code):

```

1. beginglyph(H);
2.   save serif_tl, serif_bl, serif_tr, serif_br; % local
3.   path serif_tl, serif_bl, serif_tr, serif_br; path letter_H;
4.   % fix serifs:
5.   sym_serif4'((wd.H,0), down, uc_stem, uc_serif_jut)(serif_br);
6.   sym_serif3'((wd.H,uc_height), up, uc_stem, uc_serif_jut)(serif_tr);
7.   sym_serif4'((0,uc_height), up, uc_stem, uc_serif_jut)(serif_tl);
8.   sym_serif3'((0,0), down, uc_stem, uc_serif_jut)(serif_bl);
9.   % fix bar:
10.  x1=x4=serif_tl.first.x; x2=x3=serif_br.first.x;
11.  y1=y2=1/2uc_height; y4=y3=y2+thin_stem;

```

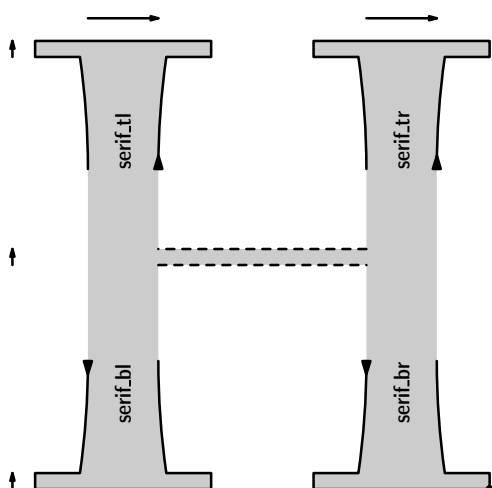


Figure 9. The letter ‘H’ from Antykwa Półtawskiego; the construction of letterforms is based on assembling “building blocks,” in this case serifs (marked with solid lines); triangles denote the starting points of the serifs, the origin of the whole outline is marked with a circle; arrows show so-called hints.

```

12. % assemble glyph:
13. letter_H = (serif_br--serif_tr--z3--z4--serif_tl--serif_bl--
14.   z1--z2--cycle) start.default;
15. Fill letter_H;
16. % final touch:
17. fix_hstem(thin_stem)(letter_H);
18. fix_vstem(uc_stem)(letter_H);
19. fix_hsbw(wd.H,marg,marg);
20. endglyph;

```

The resulting letterform is displayed in Fig. 9.

As you can see, there are several parameters involved in the code: `wd.H` (lines 5, 6, and 19), `uc_stem` (lines 5–8, and 18), `uc_serif_jut` (lines 5–8), `uc_height` (lines 6, 7 and 11), `thin_stem` (lines 11 and 17), and `marg` (line 9). Their meaning is supposed to be self-explanatory.

The macro `sym_serif` (lines 5–8) constructs a serif based on a small number of parameters: the position of a chosen construction point and direction and the sizes of the stem and jut. The result (an open path) is assigned to the last parameter (suffix). The number of the chosen construction node is passed as an optional suffix parameter (see Fig 10 and Sec. *Serifs* below); in this example, the positions of nodes 3’ and 4’ (lines 6, 8 and 5, 7, respectively) are being fixed.

Such a scheme of defining “building blocks” and assembling outlines out of them has proved very convenient. There are several macros in the Antykwa Półtawskiego package defined in this way. Note, however, that the bar of the letter ‘H’ is constructed explicitly (lines 10 and 11). It should be emphasized that it is not at all obvious which elements of a font should be classified as “building blocks” and which not (cf. Sec. *Unique features* below).

Postfix notation also proved to be handy in use in some cases. There are two examples of postfix operators in the quoted code: `first.x` (line 10) and `start.default` (line 14). The former operator returns the x -coordinate of the last node of the argument path (`serif_tl`); the latter operator shifts cyclically the numbering of nodes in such a way that the rightmost lower point of the argument path becomes the first point of the resulting

path. We will not dwell too much on this subject, as postfix notation should be regarded as “syntactic sugar” rather than as an important programming technique.

The final portion of the code of the letter ‘H’ contains three lines that deserve attention:

- Lines 17–18 specify the information about vertical and horizontal lines (stems) which have thickness to be kept uniform throughout the whole font. This information is used by PostScript interpreters for improving the process of conversion of outlines into pixels. The relevant PostScript operators controlling the process of conversion are called hints. Some programs detect hints “automagically” by finding heuristically all feasible distances. We decided that distances could be supplied explicitly without trouble, and thus only the location of hints remains to be found by a heuristic algorithm.
- Line 19 shifts the resulting glyph appropriately in the field of the character, adding left and right side bars (margins; both equal to `marg` in this case). This procedure can be regarded as a simplified version of the `adjust_fit` macro of the Computer Modern programs.

Serifs. The shape of serifs is the feature of fonts acknowledged as most characteristic. The shapes of serifs of Antykwa Półtawskiego noticeably differ from those of Computer Modern ones; the manner in which the serifs are defined and used is also different.

Let’s have a look into the code of the `sym_serif` macro, the heart of the program for the letter ‘H.’ Actually, it is an instance of a macro defining a serif with non-identical left and right juts, `asym_serif`:

```

1. vardef sym_serif@#(expr start, in_dir, stem, jut)(suffix result) =
2.   asym_serif@#(start, in_dir, stem, jut, jut)(result);
3. enddef;
4.
5. vardef asym_serif@#(expr start, in_dir, stem, l_jut, r_jut)
6.   (suffix result) =
7.   % |@#| -- if not empty, defines which point of a serif
8.   % is to be placed at |start|; otherwise |z.basic| is meant
9.   clearxy; save v_; path result; numeric result.ht, result.wd;
10.  v_=signum(ypart(in_dir));
11.  if (str @# = ""): z.basic else: z@# fi=start;
12.  % fix central part:
13.  x1-x1'=x6'-x6=v_*spread_wd;
14.  x2-x1'=if v_>0: r_jut else: -l_jut fi;
15.  x6-x5'=if v_>0: l_jut else: -r_jut fi;
16.  x2-x3=x5-x4=v_*serif_slab*slant; % the correction of slanted serifs
17.  y.basic'=y0=y7=y2-v_*spread_ht;
18.  y2=y1=y1'=y5=y6=y6'=y.basic-v_*serif_slab; y3=y4=y.basic;
19.  % fix ending points:
20.  z.basic^in_dir=z.basic';
21.  z0^in_dir=z1'^in_dir=
22.  (z.basic'+1/2stem*stem_corr(in_dir)*unitvector(in_dir rotated -90));
23.  z7^in_dir=z6'^in_dir=
24.  (z.basic'+1/2stem*stem_corr(in_dir)*unitvector(in_dir rotated 90));
25.  % define ``hook'' (alignment) points:
26.  z3'=(x2,y3); z4'=(x5,y4); % for |slant=0|, |z3=z3'| and |z4=z4'|
27.  % complete the construction:
28.  result:=z0{in_dir}..z1--z2--z3--z4--z5--z6..{-in_dir}z7;
29.  result.wd:=v_*(x3-x4); result.ht:=v_*(y3-y0);
30. enddef;

```

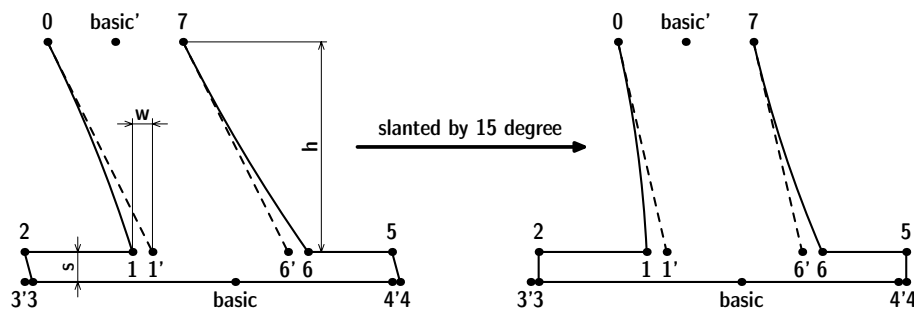


Figure 10. The construction of the serif of Antykwa Półtawskiego; the distances s , w , and h correspond to the global parameters `serif_slab`, `spread_wd`, and `spread_ht`, respectively. Observe the vertical edges of the serif *after* slanting.

Although the code looks entangled at first glance, it becomes elementary when illustrated (Fig. 10). Still, it is worthwhile to supply a few words of explanation.

- Line 11: the optional suffix parameter `@#` determines the position of a particular construction point: an empty suffix refers implicitly to `z.basic`; otherwise it is a suffix of any point used in the construction of a serif. Note that all construction points are used locally (`clearxy` in line 9).
- Lines 13 and 16–18: the global parameters used in the macro, namely, `spread_wd`, `spread_ht`, `serif_slab` and `slant`, are common to the upper- and lowercase letters, therefore they do not appear among arguments of the macro.
- Line 16: this line reflects a somewhat uncommon design decision – we assumed that the sides of the serifs should be vertical after slanting, hence an appropriate correction to the position of nodes 3 and 4 is computed. For this reason, nodes `3'` and `4'` should be used for vertical alignment prior to slanting.
- Lines 20, 21, and 22: the binary operation `a^b`, defined as `a+whatever*b`, is used in the Antykwa Półtawskiego programs instead of the phrase `whatever [a, b]`; the latter form can be used if both `a` and `b` are known, while the former requires only that the value of `b` is known.
- Lines 22 and 24: the `stem_corr` operation computes the slant correction formula for a given slope (`in_dir` in this case; cf. *Sec. Slanting: another problem in elementary geometry* above).

There exists a variant of a serif used in lowercase letters (see Fig. 11). It is constructed and used in a very similar way to the serif just described, therefore we will skip the code.

Lobes. Lobes occurring in the lowercase letters ‘b,’ ‘d,’ ‘p,’ and ‘q’ are essential for the appearance of Antykwa Półtawskiego. In particular, it is important that they are similar, although not necessarily identical: ‘b’ and ‘d’ have (probably) different lobes from that of ‘p’ and ‘q’ (Fig. 11).

The definition of lobes complies with the scheme that the programs for the construction of serifs manifest. In particular, the header of the macro `lc_lobes` resembles that of `asym_serif`:

```
vardef lc_lobes@#(expr start, width, height, raise, drop)(suffix result)
```

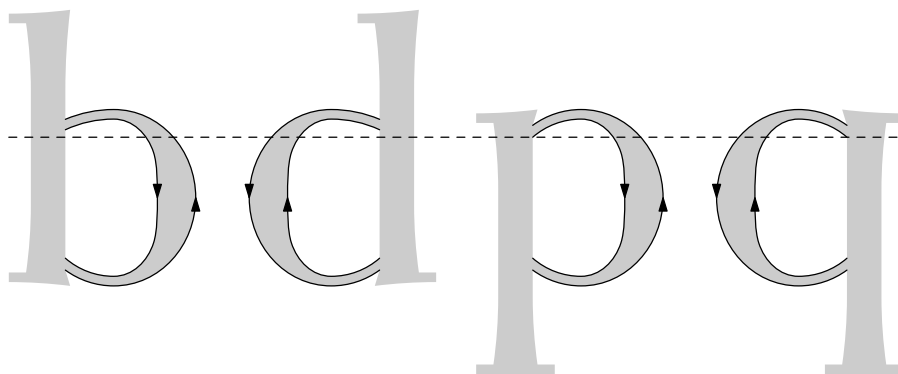


Figure 11. The construction of the lobes of Antykwa Półtawskiego (marked with a solid line). Observe a slight difference of the lobes in ‘b’ and ‘d’ as compared to the lobes in ‘p’ and ‘q’ (the drop of the upper part). The job is done by the macro `lc_lobes` which calculates both inner and outer edge appropriately oriented.

There are more kinds of lobes (letters: ‘j,’ ‘J,’ ‘m,’ ‘n,’ ‘u,’ ‘U’, etc.), but all of them can be defined using the same paradigm, which unifies and thus facilitates the task of assembling glyphs.

Unique features. Like the majority of fonts, Antykwa Półtawskiego contains many characteristic details (such as the shape of the bottom part of the letter ‘g’ – Fig. 3) appearing in only one glyph. Sometimes it is natural to consider them as an intrinsic part of the program for a given glyph, and sometimes it is reasonable to parameterize such local features by defining macros or constants for a single use.

Working on Computer Modern, Knuth apparently faced the same problem. Dozens of coefficients evidently introduced *ad hoc* can be encountered in the Computer Modern programs. For example, the program for the roman letter ‘a’ contains several “magic” numbers (the most striking ones are underlined>):

```

...
if serifs: pos1(flare,180); pos2(hair,180);
pos3(vair,90); lft xlr=hround max(u,2.1u-.5flare); x3=.5w-.5u;
y1=min(bh+.5flare+2vair+2,.9[bh,h]-.5flare);
bulb(3,2,1); % bulb
else: pos1(5/7[vair,flare],95); x1=good.x 1.5u; xlr:=good.x xlr;
pos3(1/8[vair,thin_join],90);
x3=.5w-.2u; top ylr=vround .82[bh,top y3r];
filldraw stroke term.e(3,1,left,.9,4); fi % terminal
pos4(stem,0); rt x4r=hround(w-2.5u+.5stem); y4=1/3[bh,h];
pos5(stem,0); x5=x4; y5=max(.55bh,2vair);
...

```

It seems that there is no simple answer to the question “to parameterize or not to parameterize?” – the choice is unavoidably controversial.

Documenting Antykwa Półtawskiego

Knuth apparently did not think about literate programming *in* METAFONT (although METAFONT itself is written in WEB), therefore there are neither *tangle* nor *weave* equivalents for the METAFONT language. Knuth equipped, however, the METAFONT system with a simple formatting program, MFT, that converts a METAFONT source into a format-

In general, all objects are supposed to be drawn by the endglyph macro, i.e., all drawing operations are deferred. The same concerns labelling, which necessitates the redefinition of labelling macros.	Zakłada się, że wszelkie operacje rysowania są „odraczane” i realizowane dopiero przez makro endglyph . To samo dotyczy etykietowania, skąd konieczność przeddefiniowania makr etykietujących.
---	---

```

vardef pen_labels @#(text t) =
  if project > 2: % proofing level
    forsuffixes $$ = l, r: forsuffixes $ = t:
      if known z$. $$: makelabel @#(str $$. $, z$. $$) fi;
    endfor endfor
  fi
enddef;

```

Figure 12. An example of the formatting of the documentation of Antykwa Półtawskiego: the excerpt from the documentation corresponding to the source quoted above.

ted \TeX document. MFT is smart enough to be used in a *literate* style, though obviously only for “weaving.”

The basic idea is the same as in Knuth’s WEB system: the METAFONT source is interleaved with \TeX code. It can be achieved due to a peculiar MFT convention, in that a double percent sign has dual meanings: for METAFONT it denotes just a usual comment, whereas for the MFT program it means that the text appearing after a double percent is to be inserted verbatim into the resulting \TeX document.

Obviously, the MFT program can also be used for formatting METAPOST sources. A typical fragment of the “literate documentation” of Antykwa Półtawskiego looks as follows:

```

%% \-----
%% In general, all objects are supposed to be drawn by the
%% {\bf endglyph} macro, i.e., all drawing operations are deferred.
%% The same concerns labelling, which necessitates redefinition
%% of labelling macros.
%% \-
%% Zak/lada si/e, /ze wszelkie operacje rysowania s/a ,,odraczane''
%% i~realizowane dopiero przez makro {\bf endglyph}. To samo
%% dotyczy etykietowania, sk/ad konieczno/s/c przeddefiniowania
%% makr etykietuj/acych.
%% \-----
vardef pen_labels@#(text t) =
  if project>2: % proofing level
    forsuffixes $$=l,,r: forsuffixes $=t:
      if known z$. $$: makelabel@#(str$. $$,z$. $$) fi;
    endfor endfor
  fi
enddef;

```

In this case, the \TeX code invokes the macro `\-` which typesets the bilingual description of the METAPOST source. The ensuing METAFONT code is formatted by the MFT program. The result of the typesetting process is displayed in Fig. 12.

METAPOST, like METAFONT, is capable of generating hardcopy proofs. Actually, METAPOST proofs are EPS files, therefore they can be used in a wider context than METAFONT ones. In particular, they can be included into the formatted sources. We used this technique in the Antykwa Półtawskiego sources. The examples of pages containing proof illustrations are shown in Fig 13.

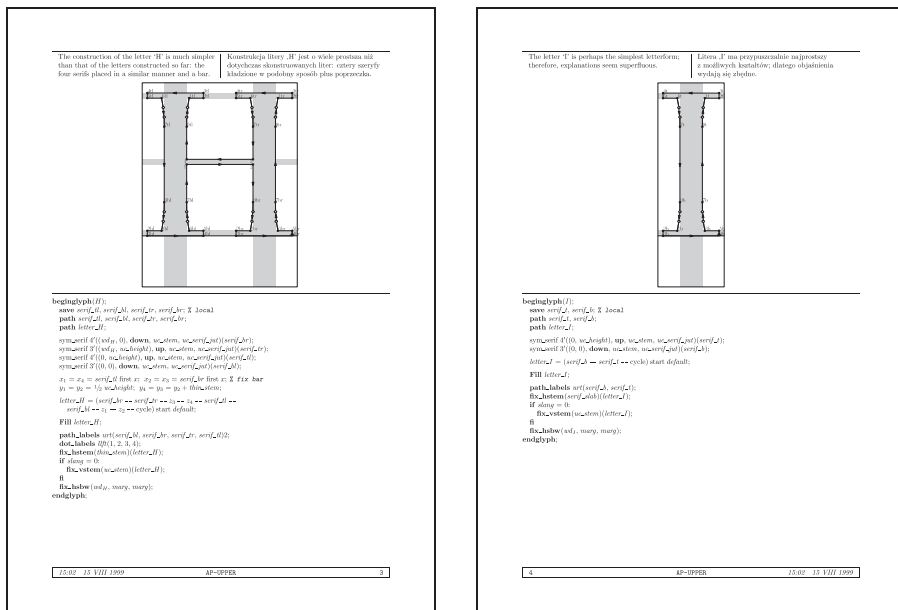


Figure 13. The appearance of the documentation of Antykwa Półtawskiego: the METAFONT source can contain references to proofs.

Postprocessing METAPOST output

The idea of postprocessing is rooted in the design principles of $\text{T}_{\text{E}}\text{X}$. There are several $\text{T}_{\text{E}}\text{X}$ -oriented utilities belonging to the standard $\text{T}_{\text{E}}\text{X}$ distribution: POOLTYPE, DVITYPE, TFTOPL, PLTOTF, VFTOVP, and VPTOVF. In a sense, all of them, as well as $\text{T}_{\text{E}}\text{X}$ drivers, can be regarded as postprocessors.

There are also METAFONT-oriented postprocessors: MFT, GFTYPE, GFTOPK, and PKTYPE (TFTOPL and PLTOTF could also be included here).

So far, there are no publicly released postprocessors for METAPOST, but sooner or later they are bound to appear, the more so as the METAPOST system is, actually, based on postprocessing (METAPOST \rightarrow $\text{T}_{\text{E}}\text{X}$ \rightarrow DVI \rightarrow METAPOST).

Knuth insisted that his WEB system was independent of external systems: he programmed all the above-mentioned utilities in PASCAL. In the early 'eighties such an attitude was plausible, but – in our humble opinion – it no longer is. Nowadays, there exist stable, public domain systems for processing text and binary files. Our experience shows that a lot can be achieved with the aid of a simple text file processor, AWK ([11]). A few years ago, it was adopted as a part of the Gnu Project (GAWK) which guarantees both stability and maintenance in the future. Since METAPOST generates well-formed 7-bit ASCII files, we decided to use GAWK for postprocessing METAPOST output.

A *Type 1* font consists of a binary file PFB (or its hexadecimal equivalent, PFA), containing the description of glyph shapes, and should be accompanied by a text metric file AFM.

In theory, GAWK might have written the hexadecimally encoded PFA files, but there is another tool, better suited for this purpose, namely, the T₁UTILS package ([13]). It contains both an assembler and a disassembler of *Type 1* fonts. Obviously, we need the assembler first of all, but the disassembler also proves useful from time to time.

Having METAPOST, GAWK and the *Type 1* assembler, we are ready to accomplish the process of generating *Type 1* fonts. It consists of three steps:

- I. METAPOST compiles the font source and writes each glyph to a separate EPS file;

additional information (e.g., hints, the name of a glyph, etc.) is written by special commands and stored as PostScript comments after the header of an EPS file.

2. GAWK processes the resulting EPS files along with a few auxiliary (configuration) files and creates an AFM file and an input file for the *Type 1* assembler.
3. The *Type 1* assembler translates the intermediate result generated by GAWK into a PFB file.

In order to use the resulting font in \TeX , one more step is needed: the installation of a font.

The main part of the task of the installation of a PostScript font is generating the relevant \TeX font metric file – TFM. This can be performed by using, e.g., a popular program AFM2TFM by Tomasz Rokicki. There are several packages facilitating the installation of PostScript fonts for \TeX , most notably *fontinst* ([12]). We decided, however, not to include any of them into our METAPOST-to-*Type 1* machinery.

The best tool for generating TFM files is of course METAFONT. At present, the sources of Antykwa Półtawskiego are unuseable by METAFONT. We are thinking about preparing a format acceptable to both METAFONT and METAPOST and adding one more step to the process of the generation of *Type 1* fonts: generating a TFM file by METAFONT from the same source. It seems that, in general, there are no fundamental obstacles, but, as wise people say, don't count your chickens before they are hatched.

Conclusions

The process of generating *Type 1* fonts turned out to be fairly efficient. It is comparable with the process of generating PK files. Therefore, it is possible to generate the font variations on the fly, in accordance with our postulates (see Sec. *Antykwa Półtawskiego*). There remains the problem of the \TeX -METAPOST interface, but that must be a matter for the future.

There is, however, a distressing side of the story: is it worthwhile to put a lot of energy into an obsolescent *Type 1* format? What about *Multiple Master*, *TrueType*, *OpenType*, and a multitude of other fonts?

Adobe PostScript is a stable worldwide standard. So far, Adobe have paid attention to the backward compatibility of PostScript versions. We hope that the situation will endure for a few years more and that *Type 1* fonts will be usable for a pretty long period.

From the point of view of \TeX applications, the quality and the functionality of *Type 1* fonts is sufficient. The advantages that can be gained by using METAPOST seem at least equivalent, if not more valuable, than the glorified features offered by the above-mentioned font formats. For example, you may regard the technique described in this paper as a \TeX -oriented implementation (and, in some respects, a generalization) of the most important features of *Multiple Master* fonts.

Knuth says in his *The METAFONT book*: *It seems clear that further work with METAFONT has the potential of producing typefaces of real beauty*. This, nobody can deny. We believe, however, that replacing the word “METAFONT” by “METAPOST” offers even better prospects.

References

- [1] Donald E. Knuth, *The METAFONT book*, Addison-Wesley, seventh printing, 1992.
- [2] Andrei Slepukhin, *private communication*, January, 1996.
- [3] Richard J. Kinch, *MetaFog: converting METAFONT shapes to contours*, TUGboat **16** (3), pp. 233–243, 1995.

- [4] Richard J. Kinch, *Belleek: A call for METAFONT revival*, Proc. of 19th Annual TUG Meeting, AuGUST 17–20, 1998, Toruń, Poland, pp. 131–136.
- [5] Taco Hoekwater, *Generating Type 1 fonts from METAFONT sources*, Proc. of 19th Annual TUG Meeting, AuGUST 17–20, 1998, Toruń, Poland, pp. 137–147.
- [6] John D. Hobby, <http://cm.bell-labs.com/who/hobby/MetaPost.html>
- [7] *Adobe Type 1 Font Format*, Addison-Wesley, 1990.
- [8] Bogusław Jackowski and Marek Ryćko, *Labyrinth of METAFONT paths in outline*, Proc. of 8th European T_EX Conference, September 26–30, 1994, Gdańsk, Poland, pp. 18–32.
- [9] Bogusław Jackowski, *A METAFONT-EPS interface*, Proc. of 9th European T_EX Conference, September 4–8, 1995, Arnhem, The Netherlands, pp. 257–271.
- [10] *ROEX: a METAFONT macro package accomplishing operations on paths, commonly known as “removing overlaps” and “expanding strokes,”*
<ftp://ftp.GUST.org.pl/TeX/graphics/MF-PS/roex/>
- [11] Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger, *The AWK Programming Language*, Addison-Wesley, 1988.
- [12] Alan Jeffrey, Rowland McDonnell, Ulrik Vieth, *Fontinst: the font installation package*, <ftp://ftp.gust.org.pl/TeX/fonts/utilities/fontinst/>
- [13] I. Lee Hetherington, *Utilities for assembling and disassembling Type 1 fonts*,
<ftp://ftp.gust.org.pl/TeX/fonts/utilities/tlutils.zip>

Acknowledgements

The authors express their hearty thanks to the Polish T_EX Users Group GUST for partially sponsoring the work on Antykwa Półtawskiego and to Phil Taylor for many valuable comments concerning the presentation of the work.

metapost

A macro routine for writing text along a path in MetaPost

Santiago Muelas
Departamento de
Mecanica
E.T.S. de Ingenieros de
Caminos C. y P. (U.P.M.)
Ciudad Universitaria.
Madrid 28040
smuelas@mecanica.upm.es
<http://w3.mecanica.upm.es/~smuelas>

abstract

In this article we show a general macro written in *pure metapost* for putting *any* text using *any* font over *any* path. The routine will be explained in detail and some graphics will be included for clarifying purposes. Very special thanks are due to **Maarten Gelderman** who has made the final translation with the biggest care and interest.

keywords

T_XP, MetaPost, L^AT_EX, METAGRAF, awk, T_EX

Introduction

In one of the steps of the construction of the program METAGRAF, we feel ourselves obliged to find a method for writing curved text. – Our goal with METAGRAF is to give T_EX/L^AT_EX users a simple and strong tool for graphics inclusion in text pages. – We knew the very simple way to achieve this using PsTricks, and at a certain moment we had the temptation to change to this language. Thanks to the very valuable advice of some important names of the *Meta World*, we decided to do it the *hard way* and we stopped the development of our program to concentrate on writing a routine general enough to be included as a general macro in MetaPost packages.

After one month of hard work to find something that could fulfill our *desiderata* we think that finally we have it. This *something* in the form of a quite simple metapost routine is what will be shown and explained in this paper.

Glyph and Boxes

As our knowledge of Metapost is limited – and we think it is so for the majority of its users – our first aim was to find a way to measure characters or, more precisely, glyphs and their corresponding boxes. As Metapost is a sibling of Metafont and knowing the very systematic way used by D. Knuth in his studies and great creations, we were sure that some way could be found because, in fact, Metapost uses T_EX for translating math text.

We looked for a hint to our problem in Hobby's User Manual, but although a last page in it shows a way to recover separate parts of a picture, this was not a big help for our problem, and the treatment of text in this Manual is extremely sober. Nevertheless, two well known capabilities of Metapost were going to form the foundation for the construction of our routine:

- The capability of finding the size of the bounding box of a picture.
- The capability of obtaining a substring from a string.

After spending a few days spend looking for some *miraculous way* of achieving our goal, it was clear that the good way was already found. The only real need was to assure that the bounding boxes had no margins – truecorners:=1 was o.k. – and that it was possible

Special thanks are due to Juan J.Arribas, Hans Hagen and Boguslaw Jackowsky

to obtain the *length* of a string – in number of characters – as in the Manual this point is unclear. After looking at a few routines in the distribution and having found a couple in which this *length* is obtained, things became more and more clear.

The big strength of MetaPost

So, the first part of our job was almost finished. We only needed to write and test it. This first part was to find the glyphs and the size of their corresponding boxes one by one. Once this is done, we will know how to *rewrite* the string with the glyph in the exact place, regarding their separation – positioning them in due vertical point was the third and last part. We will speak about the second part in a few lines.

Now, lets write the few lines needed to accomplish our first part. The steps will be:

1. Convert the string to a picture and read the total length.
2. Change the string to another without the last character.
3. Repeat the first step. The difference between both lengths will be the width of this last character.
4. Consider that the original string is the second obtained and repeat all the steps. We will obtain the width of the last glyph before the end.

We see that this is very simple to program as it is a loop. The lines of code will be:

```
% input s & pi
truecorners:=1; string s,ps; path pi; picture pic, pt;
for i = length s step -1 until 1:
  ps:=substring(0,i) of s;
  pic:=thelabel(ps, (0,0));
  long[i] = 2*(xpart urcorner pic);
endfor;
```

After this few lines, the distances from the origin of all our glyphs are stored in the array **long[]**. It is necessary to use a second loop to put every single glyph in its correct place on the path¹:

```
for i = 0 upto (length s) -1:
  sp:= substring(i,i+1) of s;
  if sp < `` ``:
    x:= (long[i] + long[i+1])/2;
    pt:= thelabel(sp, (0,0)) shifted x;
    % Here the instruction to draw the glyph <-----
    fi;
  endfor;
```

Now, we need to solve the second part of our problem. The width of our glyphs and the distances between them are known. We could put them in any place, but if we want to put them in a certain line – the path – we need to know how to put them on an arbitrary line – the path – we need to know how to do that. We need to know the length along the curve to be used and the angle that the curve forms with the coordinate axis in the different points of placement. This task looks quite hard, but we must not forget that now we are dealing with one of the strongest points of MetaPost and all that is needed and

1. We will elaborate on this second loop later.

already mentioned can be written in just one line of code². – Metapost’s math capabilities can be quite impresives. – This line is the second loop we mentioned earlier and must be included there. Here is the line:

```
draw pt shifted (-xpart pt, h) rotated angle direction arctime x
of pi of pi shifted point arctime xpart pt of pi of pi;
```

It must be said now that the variable **h** that appears in this code, has not yet been obtained. This variable will correspond to the third part of our search and represents the *height* at which the glyph is to be placed.

Working with fonts

The moment has arrived in which we have to enter the third part of our *business*. Here we need to obtain detailed knowledge of every glyph. And for this part we must say that there is little to be done with Metapost, being the discussion centered in a matter more generic than what we have already seen³.

Our goal in this part is to obtain the exact size each of the glyphs in the string that will be put along the curve. Not only the sizes of the boxes but also the position of the baseline in these boxes needs to be determined. After a couple of days looking for a way to find this data without the need of knowing the specific font used, we realized that this is not possible, so we have gone directly to the sources, that is: the study of fonts. As it does not seem to be a matter closely related to Metapost, we will not explain in detail the steps we carried, but we will go directly to the results obtained and the way the final data are used by TXP. There are some points that need be mentioned.

- Two types of files related to fonts exist and usually they can be found in any \LaTeX distribution. Those are the files related with the geometry of the boxes and the position of the baseline in them and the files where the shapes of the glyphs are stored. We are concerned with the first ones.
- The files where all the description needed usually resides are the ones with the extension *.afm. Unfortunately, these files are not updated and do not correspond with the precision needed to the real fonts used. The results obtained were just not perfect.
- Another group of files that can be used are those with extension *.pl. They contain a lot of information not needed but they have an enormous advantage over the metrics – or *.afm – ones. This advantage will be come more clear after the next group of files has been discussed.
- The files needed by the computer to be able to use a font, are those with the extension *.tfm⁴. Those files are not readable. Nevertheless, usually there is an application found on every distribution of \LaTeX called **tftopl** that converts a tfm-file to a pl-file.
- Now we can understand the advantage mentioned. Once the tfm-files have been located, we can create the corresponding *.pl files. These files are both readable and detailed. Everything needed by our macro regarding fonts is included in the *.pl files. And we can obtain these files from the same tfm-files that are used for actually writing the glyphs. That means that we can obtain an **absolute precision** with this system. So, this is the system adopted by TXP.

2. This line of code has been written by Juan J. Arribas

3. For this part of our work we have found an enormous help in the book “ \TeX UNBOUND” from *Alan Hoenig*.

4. There are also the files that contain the shapes of glyphs, but they are of little interest for us at this moment.

The pl-files need to be created once for every font. When they exist, we don't need to recreate anymore, unless the original fonts change. In the pl-files TXP will find an array that relates the number of the ASCII character with the height to place the center of the box of the glyph over the baseline. Something like: `hig[65]:=3.2576`, for the letter **A**. We will explain later on how to create those files. For the shake of understandability, let's suppose that we have already obtained the arrays.

Putting everything in Place

We can come back to our routine to arrange all our knowledge and write it in its totality – for the moment. In the second loop, the one that will put the glyph in the correct place, we leave a gap and write in it: *Here the instructions to draw the glyph*.

The method used by TXP is to obtain the ASCII number corresponding to the character that must be placed. Then, go to the array of data and look at the value corresponding to this character: the value of the variable **h**. Once obtained, the next lines of this loop can be accomplish with no problem. The corrected version of the second loop will be:

```

for i = 0 upto (length s) -1:
  sp:= substring(i,i+1) of s;
  if sp < `` ``:
    x:= (long[i] + long[i+1])/2;
    pt:= thelabel(sp, (0,0)) shifted (x,0);
    for j = 16 upto 244:
      if sp = char(j):
        k:= j;
        fi;
      endfor;
      h:=hig[k];
      draw pt shifted (-xpart pt, h) rotated angle direction
        arctime x of pi of pi shifted point arctime xpart pt
        of pi of pi;
    fi;
  endfor;

```

In the final macro the first loop will be placed in front of the beginning of this second one. Additionally, some inclusions of a general kind will be made.

Compacting it a little bit and writing the complete routine, we have:

```

1  def txp(expr s, pat, hig) =
2  picture pt, pic; string sp, ps; path pi; truecorners:= 1;
3  pi:= pat shifted(-xpart center pat, -ypart center pat);
4  for i = length s step -1 until 0:
5    ps:= substring(0,i) of s;
6    pic:= thelabel(ps, (0,0));
7    long[i] = 2*(xpart urcorner pic);
8  endfor;
9  for i=0 upto (length s) - 1:
10   sp:= substring(i,i+1) of s; if sp < " ":
11     x:= (long[i]+long[i+1])/2;
12     pt:= thelabel(sp, (0,0)) shifted (x,0);
13     for j = 16 upto 244:
14       if sp = char(j): k:=j; fi; endfor;
15     draw pt shifted (-xpart pt, alt[k]) rotated angle
        direction arctime x of pi of pi shifted point arctime

```

```

        xpart pt of pi of pi;
16    fi; endfor;
17    enddef;

```

The program – ej.mp – that will call this macro can be, for example:

```

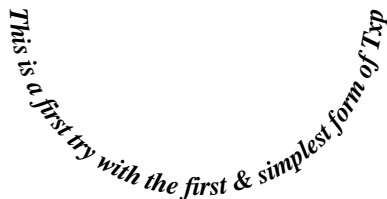
1  beginfig(10); u=0.25mm; string s; path pat;
2  input fptmbi8r; defaultfont:="ptmbi8r";
3  s="This is a first try with the first & simplest form of Txp";
4  pat:= fullcircle scaled 140 shifted(300u,500u) rotated 180;
5  txp(s,pat,alt); endfig; end

```

A few things must be said to fully understand the listing:

- The file **fptmbi8r** is the one we have created that contains the font data needed. More about that, later on.
- This file is an array from value 16 to 244, to establish the necessary coincidence of ASCII codes.
- The elements in the array are called **alt[]** and not **hig**.
- The negative sign that precedes – sometimes – the value **xpart** is a consequence of the fact that we are working – sometimes – with the center of the bounding box and we look for the left corner.

So with the sole exception of not knowing exactly what is the file **fptmbi8r** – that is totally independent from the metapost code – we have obtained a first macro to write text along a curve. In this case, a circle. Let's look the result of our small program, using the font Times Italic:



The image shows a circular arc drawn with a thin black line. Inside the arc, the text "This is a first try with the first & simplest form of Txp" is written in a black, italicized serif font. The text follows the curvature of the arc, starting from the top left and ending at the top right.

Figure 1. The first figure created with TXP

The Joy of Txp: Parameters

Although this first version of TXP works correctly it is clearly limited in its capabilities. We would like to have additional functionality like scaling the glyph or the path, being able to begin at any point in the path, placing the glyph over or under the baseline or in the middle, modifying the separation between the characters and so on.

We will show the small modifications needed to obtain these capabilities and their effect in the final image.

Firstly we will introduce the capability of scaling the glyph. This is the same as scaling the string written. This is just a matter of simply sending the scale wanted from the program to the macro as a new parameter. Once received this value becomes a new constant that can be called, for example **es**. Only some slight modifications to TXP are necessary to incorporate this new capability:

- Line 1 must take account of the new parameter.

- Line 6 of the macro must be changed from:

```
6_old    pic:= thelabel(ps, (0,0));          to:
6_new    pic:= thelabel(ps,(0,0)) scaled es;
```

- The same with line 12:

```
12_old   pt:= thelabel(sp, (0,0)) shifted (x,0);
12_new   pt:= thelabel(sp ,(0,0)) scaled es shifted (x,0);
```

- And finally the same with line 15:

```
15_old   draw pt shifted (-xpart pt, alt[k]) rotated angle
direction arctime x of pi  of pi shifted point arctime
xpart pt of pi  of pi;

15_new   draw pt shifted (-xpart pt,es}* alt[k])
rotated angle direction arctime x of pi  of pi shifted point
arctime xpart pt of pi  of pi;
```

And, also in the program, the call to the macro must include this new parameter. If we make this changes and repeat the same figure as above but increasing the scale to a value close to 2, we will obtain:

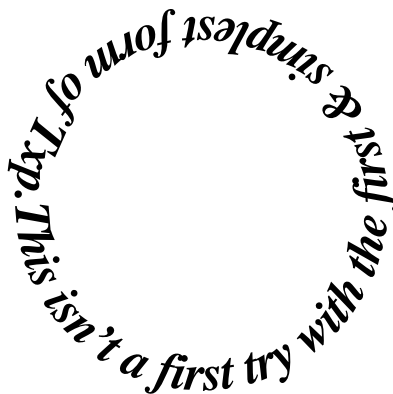


Figure 2. Scaling the glyph to the double

For scaling the path, only one *active* line of the macro must be modified, and this is line 3. In the modified version it includes the effect of this scaling, and if the factor to scale is called **ef**, this line must be written as follows:

```
3_new    pi := pat scaled ef shifted(-xpart center pat*(ef-1),
-y part center pat*(ef-1));
```

If this line is changed and the corresponding parameters included, maintaining the values given for the last figure, and giving a value to the scale of the path equal to 2, we obtain the figure:

The changes done are clearly visible and the result obtained the conform to our expectations.

In this same way and philosophy it is possible to add many new features to the macro, but to keep the size of this paper reasonable we will not continue in this *step by step* way of increasing the capabilities. At the end we will write a quite complete version of TXP

This isn't a first try with the first & simplest form of TXP.

Figure 3. Scaling the path to the double

and an example that will show the many capabilities are implemented⁵

Finishing the fonts discussion

At this stage of this paper, only one point has remained in what could be called *the mysterious depth*. We are referring to the font data files. A big part of the mystery has already been explained and we have left for the last part the total knowledge because as we said before, this really is not a metapost affair.

What TXP needs is the possibility to access certain data related with the font geometry. More specifically, the size of the bounding box of every glyph and its position in relation with the baseline. All those data are included in binary form in the files *.tfm as we have said. When we transform these files to the format *.pl, we make readable the content of the .tfm file. So, the file .pl contains all that is needed by TXP. The only thing that remains for giving the data to TXP is just to organize and arrange them. But Metapost is not an ideal language for doing that and a small helping programming language, like AWK is the perfect one. So, we have written a few lines in awk that we wish to show now. We also will explain the way to use this very short routine.

The routine gentypl.awk that we are going to describe does a simple but fundamental job. GENTYP looks at all the lines of the .pl file, reads some values from the lines that begin with the words: CHARACTER, CHARHT and/or CHARDP, makes a simple calculation and writes the result to an auxiliary file called **fontdat**. If we have taken as our .pl file the one corresponding to the font “palatino bold roman”, its name would be **pplb8r**. Once we have obtained the file fontdat we only need to change this name to **fpplb8r**, and this is the file to that will be used by TXP. For any other font, the way of proceeding would be exactly the same IF the font is of the type “8”, so up to 255 possible glyph.

Let's write the listing of this help routine. Here is gentypl.awk:

```
BEGIN {i=0}
  $1 == "(CHARACTER" { letter[i]=$3; i++}
  $1 == "(CHARHT" { high[i-1]=10*$3;}
  $1 == "(CHARDP" { deph[i-1] = 10*$3 ;}
END{for(j=0;j<229;j++) print "alt["j+15"]:= \
```

5. We made the conscious decision to limit the capabilities to a reasonable amount of parameters, but they can be increased with no problem. For example, to add the possibility of scaling the text with a different horizontal and vertical scale is trivial, and the same is true for the shearing of the bounding boxes that can furnish an interesting tool for special cases in which it is desired to do something *fantaisiste* as, for example, transforming an italic font in a vertical one or the contrary.

```
"(high[j]+deph[j])/2-deph[j]  "fontdat";}
```

Summarizing, what is needed to create the font data file is just to write:

```
awk -f gentyp.awk pplb8r.pl and then:
mv fontdat fpplb8r
```

And that's all. Once this preparatory work is done for all the fonts usually employed, there is no need to care anymore.⁶

The best at the end

In what follows we will show the final aspect of the `txp.mp` macro and an example of an application with different fonts, scales and colors. We will finish by giving a few recommendations of a more practical nature.

Final Listing Of Macro `txp.mp`.-

```
def txp(expr s, pat, es, ef, hi, tr, se, lc, th, hig) =
picture pt ,pic; string sp, ps; color loc; path pi;
truecorners:=1;
pi := pat scaled ef shifted(-xpart center pat*(ef-1), \
-y part center pat*(ef-1));
long[0]:=tr;
for i = length s step -1 until 1:
  ps:=substring(0,i) of s;
  pic:=thelabel(ps,(0,0)) scaled es;
  long[i]:=((2+se)*(xpart urcorner pic) + tr);
endfor;
for i=0 upto (length s) - 1:
  sp:= substring(i,i+1) of s;
  if sp < " ":
    x:= (long[i]+long[i+1])/2;
    pt:= thelabel(sp ,(0,0)) scaled es shifted (x,0);
    for j = 16 upto 244:
      if sp = char(j): k:=j; fi; endfor;
      h:=es*(alt[k] + hi);
      draw pt shifted (-xpart pt,h) rotated angle direction \
arctime x of pi of pi shifted point arctime xpart pt of pi \
of pi withcolor lc;
      fi; endfor;
  if th < 0:
    pickup pencircle scaled th;
    draw pi withcolor red;
  fi;
enddef;
```

With this macro, there are practically no limitations to the capability of writing text strings of any length – although they may not be longer than a single paragraph – using the type and size of fonts and the shape of the path desired. It is possible to mix all those components in a single program as will be shown in the example that follows.

6. Included with this paper will be the fontdat files of three very important families of fonts: Times, Palatino and Helvetica. We also will include the file corresponding to ZapfChancery.

An application to finish with

The program that will be shown immediately gives a quite interesting idea of the possibilities of TXP. Here is the listing:

```

beginfig(1);
input txp; u=0.25mm; color loc; string s; path a;
es:=1; ef:=1; hy:=0; tt:=0; sep:=0; loc:=black; lin=0;

input fptmbi8r;
defaultfont:="ptmbi8r";
s:="WRITING ON THE PATH IS AMUSING AND EASY...WRITING ON THE PATH \
IS AMUSING AND EASY...WRITING ON THE PATH IS AMUSING AND EASY... \
WRITING ON THE PATH IS AMUSING AND EASY...WRITING ON THE \
PATH IS AMUSING AND EASY...";
a:=(192u,768.0u).. controls (124u,824.0u) and (128u,904.0u).. \
(166u,934.0u).. controls (204u,964.0u) and (268u,944.0u).. \
(320u,886.0u).. controls (372u,828.0u) and (380u,768.0u).. \
(438u,742.0u).. controls (496u,716.0u) and (584u,752.0u).. \
(590u,832.0u).. controls (596u,912.0u) and (548u,952.0u).. \
(504u,974.0u).. controls (460u,996.0u) and (348u,1004.0u).. \
(294u,938.0u).. controls (240u,872.0u) and (260u,776.0u).. \
(330u,732.0u).. controls (400u,688.0u) and (660u,676.0u).. \
(688u,860.0u);
hy:=-3;sep:=0.2;loc:=blue;
txp(s,a,es,ef,hy,tt,sep,loc,lin,alt);

input fpzcmi8r;
defaultfont:="pzcmi8r";
s:="YES !!";
a:=(400u,850u)--(500u,850u);
es:=4;hy:=0.;sep:=0.;loc:=red;
txp(s,a,es,ef,hy,tt,sep,loc,lin,alt);

input fphvb8r;
defaultfont:="phvb8r";
s:"Although sometimes, it can be cumbersome.";
a:=(600u,1000u)..(425u,1100u)..(250u,1000u);
es:=1.3; sep:=.0; loc:=black;
txp(s,a,es,ef,hy,tt,sep,loc,lin,alt);

input fpplbo8r;
defaultfont:="pplbo8r";
s:"But, anyway, I like it !! ...";
a:=(250u,600u)--(600u,600u);
es:=1.8; sep:=0.3;loc:=green;
txp(s,a,es,ef,hy,tt,sep,loc,lin,alt);

input fphvb8r;
defaultfont:="phvb8r";
s:":-)";
a:=(400u,500u)--(400u,400u);
es:=4;hy:=0.;sep:=0.;loc:=red;
txp(s,a,es,ef,hy,tt,sep,loc,lin,alt);

```

```

input fphvr8r;
defaultfont:="phvr8r";
s:="0";
a:=(355u,500u)--(390u,500u);
es:=11;hy:=-5.4;sep:=0.;loc:=blue;
txp(s,a,es,ef,hy,tt,sep,loc,lin,alt);
endfig; end

```

And here – in the next page – is the result of running it.

We will finish with a remark on the meaning of the parameters used in the general macro, although, as we have already said, this is just a *limited* version.

Parameter and meaning.-

```

s ---> String to write;
a ---> Path to write to;
es ---> Scale used for the fonts. (Default: 1; No scaled)
ef ---> Scale used for the path. (Default: 1; No scaled)
hy ---> Vertical placement over the path.
      (Default: 0; The path is the baseline)
tt ---> Distance between the beginning of the written
      string and the beginning of the path. (Default: 0)
sep---> Extra separation between the glyph. (Default: 0)
loc---> Color to use for each string. (Default: black)
lin---> Thickness used to draw the path. (Default: 0: No draw)

```

Note. For using Computer Modern fonts of type “7”, some small arrangements must be made. Once understood this paper, it is a very simple matter. If in doubt, visit the Web Page at <http://w3.mecanica.upm.es/metapost> or contact me directly by email; see the start of the article.



But, anyway, I like it !! ...

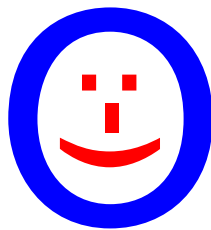


Figure 4. The last figure of this paper

metapost

MetaFun

Chapter 3: Embedded graphics

keywords

METAPOST, MetaFun, ConTEXt, graphics

abstract

This article is a nearly 100% copy of chapter 3 of the **MetaFun** manual. This chapter discusses a few alternative ways to define and include METAPOST graphics in a document.

This article contains some colors, so if you want to get the real picture, you should fetch the **MetaFun** manual from www.pragma-ade.com (beta manual download page). The **MetaFun** macros (METAPOST as well as TEX) are part of the regular ConTEXt distribution.

Introduction

This is chapter 3 of the METAFUN manual and introduces the basic definition and inclusion macros. In the previous chapters, METAPOST was introduced, while the following chapters discuss how to enhance the layout, how to achieve special effects, and how to define and debug graphics. The METAFUN manual fully covers METAPOST and the way it can be used in TEX.

At the time of this writing, the METAFUN manual spans some 260 pages. The manual is available in two versions: a reduced A4 format, and a version designed for reading from computer screens. The manual is not yet finished, but can be fetched from the CONTEXT beta manuals download page at www.pragma-ade.com or one of its mirrors.

Getting started

From now on, we will assume that you have CONTEXT running on your platform. Since PDF has full graphics support, we also assume that you use PDFTEX, or know how to go from DVI to PDF. Since this document is not meant as a CONTEXT tutorial, we will limit this introduction to the basics needed to run the examples.

A simple document looks like:

```
\starttext
  Some text.
\stoptext
```

You can process this document with the PERL based command line interface to CONTEXT. If the source code is embedded in the file `mytext.tex`, you can say:

```
texexec --pdf mytext
```

As an alternative to `-pdf`, you can explicitly set the output driver in your document:

```
\setupoutput[pdftex]
\starttext
  Some text and/or graphics.
\stoptext
```

Yet another alternative is:

```
% interface=english output=pdfTeX
\starttext
  Some text and/or graphics.
\stoptext
```

Here the interface directive tells \TeX EXEC that it should force the english user interface.

We will use color, and since traditionally \TeX is rather unaware of color, this feature is turned off by default, so, if you want to see color, you should type:

```
\setupcolors[state=start]
\starttext
  Some \color[blue]{text} and/or \color[green]{graphics}.
\stoptext
```

As an alternative, you can run \TeX EXEC like:

```
texexec --pdf --color mytext
```

In later chapters we will occasionally see some more $\text{CON}\TeX\text{T}$ commands show up. If you want to know more about what $\text{CON}\TeX\text{T}$ can do for you, we recommend the beginners manual and the reference manual, as well as the manual that comes with \TeX EXEC.

External graphics

Since \TeX has no graphic capabilities built in, a graphic is referred to as an external figure. A METAPOST graphic often has a number as suffix, so embedding such a graphic is done by:

```
\externalfigure[graphic.123][width=4cm]
```

An alternative method is to separate the definition from the inclusion. An example of a definition is:

```
\useexternalfigure[pentastar][star.803][height=4cm]
\useexternalfigure[octostar][star.804][pentastar]
```

Here, the second definition inherits the characteristics from the first one. These graphics can be summoned like:

```
\placefigure
  {A five||point star drawn by \METAPOST.}
  {\externalfigure[pentastar]}
```

Here the stars are defined as stand-alone graphics, in a file called `star.mp`. Such a file can look like:

```
def star (expr size, n, pos) =
  for a=0 step 360/n until round(360*(1-1/n)) :
    draw (origin -- (size/2,0)
      rotatedaround (origin,a) shifted pos ;
  endfor ;
enddef ;

beginfig(803) ;
  pickup pencircle scaled 2mm ; star(2cm,5,origin) ;
endfig ;

beginfig(804) ;
  pickup pencircle scaled 1mm ; star(1cm,8,origin) ;
  pickup pencircle scaled 2mm ; star(2cm,7,(3cm,0)) ;
endfig ;
```

end.

This `star` macro will produce graphics like:



Integrated graphics

An integrated graphic is defined in the document source or in a style definition file. The most primitive way of doing this is beginning with the definition of the graphic.

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

Next the graphic can be loaded, using:

```
\loadcurrentMPgraphic{optional setups}
```

Finally, the graphic is placed in the document with:

```
\placeMPgraphic
```

The optional setups are passed on to the figure inclusion macro, which in `CONTEXT` is the command `\externalfigure`.

Since every definition replaces the previous one, this method forces you to embed the definitions in the running text. In this document we also generate graphic while we finish a page, so there is a good change that when we have constructed a graphic which will be called the next page, the wrong graphic is placed.

Therefore you may as well forget these commands, since there are more convenient ways of defining and using graphics, which have the added advantage that you can pre-define multiple graphics, thereby separating the definitions from the usage.

The first alternative is a *usable* graphic. Such a graphic is calculated anew each time it is used. An example of a usable graphic is:

```
\startuseMPgraphic{name}
  fill fullcircle scaled 200pt withcolor .625yellow ;
\stopuseMPgraphic
```

When you put this definition in the preamble of your document, you can place this graphic anywhere in the file, saying:

```
\useMPgraphic{name}
```

As said, this graphic is calculated each time it is placed, which can be time consuming. Apart from the time aspect, this also means that the graphic itself is incorporated many times. Therefore, for graphics that don't change, `CONTEXT` provides *reusable* graphics:

```
\startreusableMPgraphic{name}
  fill fullcircle scaled 200pt withcolor .625yellow;
\stopreusableMPgraphic
```

This definition is accompanied by:

```
\reuseMPgraphic{name}
```

Imagine that we use a graphic as a background for a button. We can create a unique and reusable graphic by saying:


```
\def\MyGraphic%
  {\startreusableMPgraphic{name:\overlaywidth:\overlayheight}
   path p ; p := unitsquare
   xscaled \overlaywidth yscaled \overlayheight ;
   fill p withcolor .625yellow ;
   draw p withcolor .625red ;
   \stopreusableMPgraphic
   \reuseMPgraphic{name:\overlaywidth:\overlayheight}}
```

After this we can say:

```
\defineoverlay[my graphic][\MyGraphic]
\button[background=my graphic,frame=off]{Go Home}[firstpage]
```

Say that we have a 30pt by 20pt button, then the identifier will be name:30pt:20pt. Different dimensions will lead to other identifiers, so this sort of makes the graphics unique.

We can bypass the ugly looking `\def` by using a third class of embedded graphics, the *unique* graphics.

```
\startuniqueMPgraphic{name}
  path p ; p := unitsquare
  xscaled \overlaywidth yscaled \overlayheight ;
  fill p withcolor .625yellow ;
  draw p withcolor .625red ;
\stopuniqueMPgraphic
```

Now we can say:

```
\defineoverlay[my graphic][\uniqueMPgraphic{name}]
\button[background=my graphic,frame=off]{Go Home}[firstpage]
```

You may wonder why unique graphics are needed when a single graphic might be used multiple times by scaling it to fit the situation. Since a unique graphic is calculated for each distinctive case, we can be sure that the current circumstances are taken into account. Also, scaling would result in incomparable graphics. Consider the following definition:

```
\startMPgraphic
  draw unitsquare
  xscaled 5cm yscaled 1cm
  withpen pencircle scaled 2mm
  withcolor .625red ;
\stopMPgraphic
```

Since we reuse the graphic, the dimensions are sort of fixed, and because the graphic is calculated once, scaling it will result in incompatible line widths.



These graphics were placed with:

```
\hbox \bgroup
  \loadcurrentMPgraphic{width=5cm,height=1cm}\placeMPgraphic \quad
  \loadcurrentMPgraphic{width=8cm,height=1cm}\placeMPgraphic \egroup
```

Imagine what happens when we add some buttons to an interactive document without taking care of this side effect. All the frames would look different. Consider the following example.

```

\startuniqueMPgraphic{right or wrong}
  pickup pencircle scaled .075 ;
  fill unitsquare withcolor .8white ;
  draw unitsquare withcolor .625red ;
  currentpicture := currentpicture
    xscaled \overlaywidth yscaled \overlayheight ;
\stopuniqueMPgraphic

```

Let's define this graphic as a background to some buttons.

```

\defineoverlay[button][\uniqueMPgraphic{right or wrong}]
\setupbuttons[background=button,frame=off]

\hbox
  {\button {previous}           [previouspage]\quad
   \button {next}              [nextpage]\quad
   \button {index}            [index]\quad
   \button {table of contents} [content]}

```

The buttons will look like:



Compare these with:



Here the graphic was defined as:

```

\startuniqueMPgraphic{wrong or right}
  pickup pencircle scaled 3pt ;
  path p ; p := unitsquare
    xscaled \overlaywidth yscaled \overlayheight ;
  fill p withcolor .8white ;
  draw p withcolor .625red ;
\stopuniqueMPgraphic

```

The last class of embedded graphics are the *runtime* graphics. When a company logo is defined in a separate file `mylogos.mp`, you can run this file by saying:

```

\startMPrun
  input mylogos ;
\stopMPrun

```

The source for the logo is stored in a file named `mylogos.mp`.

```

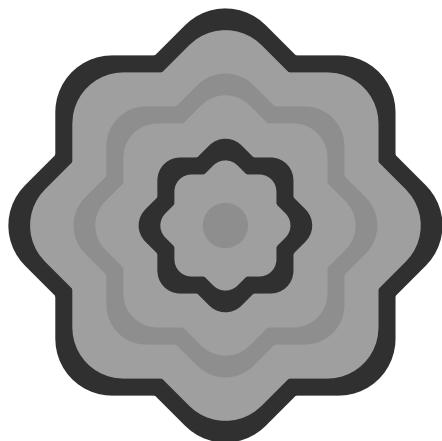
beginfig(21) ;
  draw fullsquare           withcolor .625red ;
  draw fullsquare rotated 45 withcolor .625red ;
  picture cp ; cp := currentpicture ;
  def copy = addto currentpicture also cp enddef ;
  copy scaled .9 withcolor .625white ;
  copy scaled .7 withcolor .625yellow ;
  copy scaled .6 withcolor .625white ;
  copy scaled .4 withcolor .625red ;
  copy scaled .3 withcolor .625white ;
  fill fullcircle scaled .2 withcolor .625yellow ;
  currentpicture := currentpicture scaled 50 ;

```

```
endfig ;
end .
```

In this example the result is available in the file `mprun.21`. This file can be included in the normal way, using:

```
\externalfile[mprun.21] [width=5cm]
```



Figuur 1 The logo is defined in the file `mylogos.mp` as figure 21 and processed by means of the `mprun` method.

Graphic buffers

In addition to the macros defined in the previous section, you can use `CONTEX`T's buffers to handle graphics. This can be handy when making documentation, so it makes sense to spend a few words on them.

A buffer is a container for content that is to be (re)used later on. The main reason for their existence is that they were needed for typesetting manuals and articles on `TEX`. By putting the code snippets in buffers, we don't have to key in the code twice, since we can either show the code of buffers verbatim, or process the code as part of the text flow. This means that the risk of mismatch between the code shown and the typeset text is minimized.

```
\startbuffer
You are reading the \METAFUN\ manual.
\stopbuffer
```

This buffer can be typeset verbatim using `\typebuffer` and processed using `\haalbuffer`, as we will do now:

An other advantage of using buffers, is that they help you keeping the document source clean. In a many places in this manual we put table or figure definitions in a buffer and pass the buffer to another command, like:

```
\placefigure{A very big table}{\haalbuffer}
```

Sometimes it makes sense to collect buffers in separate files. In that case we give them names.

This time we should say `\typebuffer[mfun]` to typeset the code verbatim. Instead of `TEX` code, we can put `METAPOST` definitions in buffers.

Buffers can be used to stepwise build graphic. By putting code in multiple buffers, you can selectively process this code.

```

\startbuffer[red]
drawoptions(withcolor .625red) ;
\stopbuffer

\startbuffer[yellow]
drawoptions(withcolor .625yellow) ;
\stopbuffer

```

We can now include the same graphic in two colors by simply using different buffers. This time we use the special command `\processMPbuffer`, since `\haalbuffer` will type-set the code fragment, which is not what we want.

```

\startregelcorrectie[blanko]
\processMPbuffer[red,graphic]
\stopregelcorrectie

```

The line correction macros take care of proper spacing around the graphic. The `[blanko]` directive tells `CONTEXT` to add more space before and after the graphic.

```

\startregelcorrectie[blanko]
\processMPbuffer[yellow,graphic]
\stopregelcorrectie

```

Which mechanism you use, (multiple) buffers or (re)usable graphics, depends on your preferences. Buffers are slower but don't take memory, while (re)usable graphics are stored in memory which means that they are accessed faster.

Communicating color

Now that color has moved to the desktop, even simple documents have become more colorful, so we need a way to consistently apply color to text as well as graphics. In `CONTEXT`, colors are called by name.

The next definitions demonstrate that we can define a color using different color models, RGB or CMYK. Depending on the configuration, `CONTEXT` will convert one color system to the other, RGB to CMYK, or vice versa. The full repertoire of color components that can be set is as follows.

```

\definecolor[color one] [r=.1, g=.2, b=.3]
\definecolor[color two] [c=.4, m=.5, y=.6, k=.7]
\definecolor[color three] [s=.8]

```

The numbers are limited to the range 0..1 and represent percentages. Black is represented by:

```

\definecolor[black 1] [r=0, g=0, b=0]
\definecolor[black 2] [c=0, m=0, y=0, k=1]
\definecolor[black 3] [s=0]

```

Predefined colors are passed to `METAPOST` graphics via the `\MPcolor`. First we define some colors.

```

\definecolor[darkyellow] [y=.625] % a CMYK color
\definecolor[darkred] [r=.625] % a RGB color
\definecolor[darkgray] [s=.625] % a gray scale

```

These are the colors we used in this document. The next example uses two of them.

```
\startuseMPgraphic{color demo}
  pickup pencircle scaled 1mm ;
  path p ; p := fullcircle xscaled 10cm yscaled 1cm ;
  fill p withcolor \MPcolor{darkgray} ;
  draw p withcolor \MPcolor{darkred} ;
\stopuseMPgraphic

\useMPgraphic{color demo}
```

The previous example uses a pure RGB red shade, combined with a gray fill.



Since METAPOST does not support the CMYK color space and native gray scales — although gray colors are reduced to the more efficient POSTSCRIPT `setgray` operators in the output— the macro `\MPcolor` takes care of the translation from CMYK to RGB as well as gray to RGB. However, there is a fundamental difference between a yellow as defined in CONTEXt using CMYK and an RGB yellow in METAPOST.

```
\definecolor[cmyellow] [y=1]
\definecolor[rgbyellow] [r=1,g=1]

\definecolor[cmdarkyellow] [y=.625]
\definecolor[rgbdarkyellow] [r=.625,g=.625]
```

Figure 2 demonstrates what happens when we multiply colors by a factor. Since we are not dealing with real CMYK colors, multiplication gives different results for CMYK colors passed as `\MPcolor`.

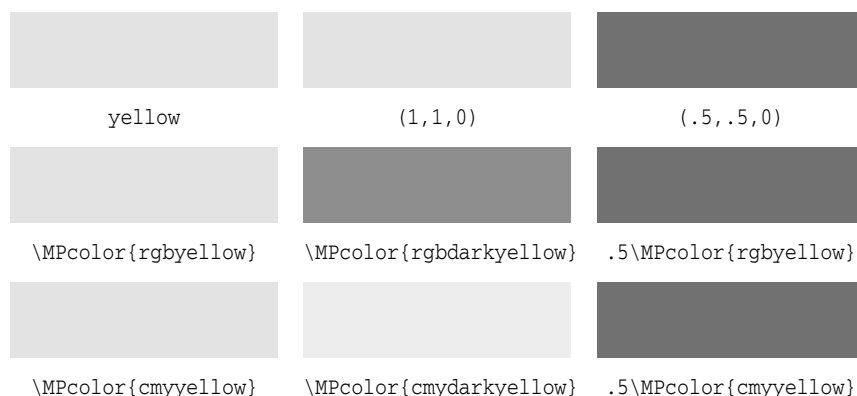


Figure 2 All kinds of yellow.

So, `.625red` is the same as `[r=.5]`, but `.625yellow` is not the same as `[y=.5]`, but matches `[r=.5,g=.5]`. Figure 3 shows the pure and half reds.

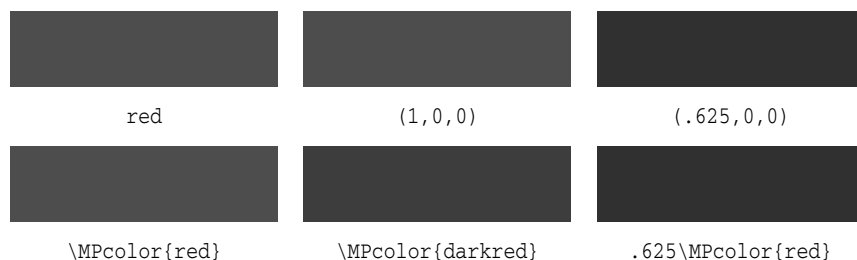


Figure 3 Some kinds of red.

In order to prevent problems, we advise you to stick to RGB color specifications when possible. That way you prevent not only conversion problems, but the also (often obscure) ways printing and viewing devices handle CMYK.

Common definitions

When using many graphics, there is a chance that they share common definitions. Such shared components can be defined by:

```
\startMPinclusions
  color mycolor ; mycolor := .625red ;
\stopMPinclusions
```

All METAPOST graphics defined in the document end up in the files `mpgraph.mp` and `mprun.mp`. When processed, they produce (sometimes many) graphic files. When using T_EX_{EXEC} to process documents, these two files are processed automatically after a run so that in a next run, the right graphics are available.

When you are using the `web2c` distribution, CON_TE_XT can call METAPOST at runtime and thereby use the right graphics instantaneously. In order to use this feature, you have to enable `\write18` in the file `texmf.cnf`. Also, in the file `cont-sys.tex`, that holds local preferences, or in the document source, you should say:

```
\runMPgraphicstrue
```

This enables runtime generation of graphics using the low level T_EX command `\write18`. First make sure that your local brand of T_EX supports this feature. A simple test is making a T_EX file with the following line:

```
\immediate\write18{echo It works}
```

If this fails, you should consult the manual that comes with your system, locate an expert or ask around on the CON_TE_XT mailing list. Of course you can also decide to let T_EX_{EXEC} take care of processing the graphics afterwards. This has the advantage of being faster but has the disadvantage that you need additional T_EX runs.

If you generate the graphics at run time, you should consider to turn on graphic slot recycling, which means that you often end up with fewer intermediate files:

```
\recycleMPslotstrue
```

There are a few more low level switches and features, but these go beyond the purpose of this manual. Some of these features, like the option to add tokens to `\everyMPgraphic` are for experts only, and fooling around with them can interfere with existing features.

One page graphics

Although all of what is demonstrated in this document is done in CON_TE_XT, some of the features discussed here can also be done in plain T_EX. In the META_FUN distribution there is a file called `plainfun.tex`, which loads the appropriate CON_TE_XT modules.

Many low level macros are rather generic, and can be used in plain T_EX without problems. However, the big advantage of using CON_TE_XT is, that graphics can be part of the text flow and that you can put them on layers. If you don't want this, and only want to make stand alone graphics, you may still consider using CON_TE_XT for that purpose.

Another advantage is that when using CON_TE_XT you don't have to bother about specials, font inclusion and all those nasty things that can spoil a good day. An example of such a graphic is the file `mfun-888` that resides on the computer of the author.

```
[file mfun-888 bestaat niet]
```

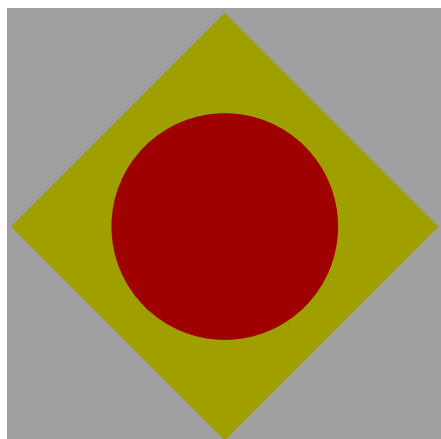
Given that the `CONTEXT` english interface format is present on your system, you can process this file with `TEXEEXEC`, for instance using `PDFTEX`. The `-once` directive saves some runtime.

```
texexec --once --pdf mfun-888
```

You can define many graphics in one file. The `TEXEEXEC` manual describes how to selectively process pages. If you use `PDFTEX`, you can include individual pages from `PDF` files:

```
\placefigure
  {A silly figure, demonstrating that stand|alone|graphics
   can be made.}
  {\externalfigure[mfun-888][page=1]}
```

In this case the `page=1` specification is not really needed. You can scale and manipulate the figure in any way supported by the macro package that you use.



Figuur 4 A silly figure, demonstrating that stand-alone-graphics can be made.

Managing resources

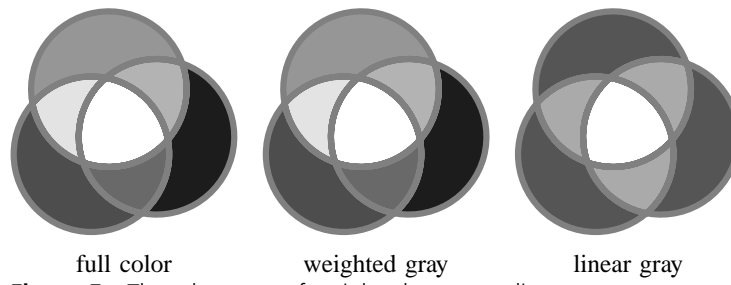
A graphic consists of curves, either or not filled with a given color. A graphic can also include text, which means that fonts are used. Finally a graphic can have special effects, like a shaded fill. Colors, fonts and special effects go under the name resources, since they may demand special care or support from the viewing or printing device.

When fonts are used, a `METAPOST` file is not self contained. This means that the postprocessing program has to deal with the fonts. In `CONTEXT`, the special driver—and `PDFTEX` support is considered as such—takes care of this. We will discuss text related issues in chapter ??.

Special effects, like shading, are supported by dedicated `METAPOST` modules. These are included in the `CONTEXT` distribution and will be discussed later in chapter ??.

Since `METAPOST` supports color, an embedded graphic can be rather colorful. However, when color support is disabled or set up to convert colors to gray scales, `CONTEXT` will convert the colors in the graphic to gray scales.

You may wonder what the advantage is of weighted gray conversion. Figure 5 shows the difference between natural colors, weighted gray scaled and straightforward, non-weighted, gray scales.



full color weighted gray linear gray
Figuur 5 The advantage of weighted gray over linear gray.

When we convert color to gray, we use the following formula. This kind of conversion also takes place in black and white televisions.

$$G = .30r + .59g + .11b$$



Kluwer Academic Publishers (KAP) is an internationally operating company, publishing 700 journals and 800 books per year. For our rapidly developing electronic publishing activities we have a vacancy for a

(La)TeX Developer

In this technically challenging position you will be responsible for various (La)TeX-related development and production activities, and for the development and implementation of software solutions at the front and back end of the production process of scientific information. The (La)TeX developer will be responsible for bridging the gap between SGML/XML content and (La)TeX-based material.

KAP is looking for a team player with, besides (La)TeX expertise, a broad technical knowledge. You should be able to demonstrate knowledge in SGML/XML/HTML, databases and programming languages such as Java, Perl, OmniMark, etc. Experience in working on different computing platforms and operating systems would be an advantage. You should be able to communicate your ideas to other members of the team and support both internal and external users, and be a champion in the production department for new technology.

Kluwer Academic Publishers, with offices also in the USA (Boston and New York) is part of the Wolters Kluwer organisation, and offers a challenging position with good conditions and career opportunities in an excellent working environment.

Send your CV and a covering letter explaining why you are the right person for this position, in English, to

Mr. Rob Doornebal
Kluwer Academic Publishers
Achterom 119
3311 KB Dordrecht

For more information on this function you can contact Mr. Rob de Jeu at (0)78 6392524 (phone) or Rob.deJeu@wkap.nl (email).