
Nederlandstalige T_EX Gebruikersgroep

MAPS: M_inutes and AP_pendiceS#12 (94.1)

Verslag:

1.	Opening 12 ^e NTG bijeenkomst 18 november 1993	1
2.	Verslag NTG bijeenkomst van 10 juni 1993	1
3.	Ingekomen stukken en Mededelingen	1
4.	Begroting 1994	2
5.	Wat verder ter tafel komt	2
6.	L ^A T _E X3	2
7.	MAPS	3
8.	Verslag bijeenkomsten	3
9.	Verslag werkgroepen	3
10.	Rondvraag	4
11.	Voordrachten: '(L ^A)T _E X gebruikersomgeving'	5
12.	Volgende bijeenkomsten	5
13.	Sluiting	5

Bijlagen:

A	Het weten waard	6
B	Van de Voorzitter	8
C	Van uw MAPS Editor	11
D	Jaarverslag NTG 1993	13
E	Financieel verslag NTG 1993	15
F	Use of T _E X and L ^A T _E X within the NTG community	17
G	The making of the 4allT _E X CD-ROM	19
H	FGBBS: de jonge jarige	22
I	4T _E X 3.20 a technical note	25
J	Frequently Asked Questions about 4T _E X	27
K	T _E X zonder omhaal; voor Atari ST en andere PC's (III)	31
L	T _E X zonder omhaal; voor Atari ST en andere PC's (IV)	34
M	L ^A T _E X2 _ε , an overview	37
N	Standard Document Classes and Packages	47
O	Upgrading old styles	50
P	A brief guide to T _E X assistants	53
Q	Building virtual fonts with 'fontinst'	58
R	Text to Hypertext Conversion with L ^A T _E X2HTML	60
S	Colour slides with L ^A T _E X and 'seminar.sty'	63
T	Making T _E X work	69
U	Review of 'Making T _E X work'	92
V	Typesetting Greek texts by T _E X and L ^A T _E X	93
W	Springer authors do it with T _E X	97
X	T _E X-verwerking bij PRAGMA	100
Y	Computerondersteund tekstgebruik	103
Z	Data publishing	106
A	Table design	108
B	BLUe's Transparencies	111
C	BLUe's Bibliography	115
D	BLUe's Verbatim	116
E	TUGboat BLUes	119
F	Book review: 'Practical SGML' & 'Math into T _E X'	151
G	The T _E X Hierarchy	153
H	Obtaining T _E X	155
I	L ^A T _E X2 _ε Second Test Release	157
J	Literate Programming Frequently Asked Questions	159
K	Report BachoT _E X '94	165
L	Report NT _E X '94	167
M	Announcement TUG '94	168
N	Announcement EuroT _E X '94	170
O	Announcement 4allT _E X CD-ROM	171
P	Cursusaankondiging 4T _E X	175
Q	Table of Contents TUGboat	176

De NTG vereniging

Voorzitter:	C.G. van der Laan, Internet: cgl@risc1.rug.nl
Secretaris:	G.J.H. van Nes, ECN, Afdeling Informatica, Petten. Internet: vannes@ecn.nl
Penningmeester:	J.L. Braams, PTT Dr Neher Laboratorium, Leidschendam. Internet: j.l.braams@research.ptt.nl
Bestuursleden:	E.H.M. Frambach, RUG, Groningen. Internet: e.h.m.frambach@eco.rug.nl J.J. Winnink. Internet: jos.winnink@cpb.nl
Postadres:	Nederlandstalige T _E X Gebruikersgroep, Postbus 394, 1740 AJ Schagen.
Postgiro:	1306238, 000-1662209-17, t.n.v. Penningmeester NTG, t.n.v. Ph. Vanoverbeke (NTG) Zoetermeer. Langenhoekstraat 28, B-8210 Veldegem
E-mail bestuur:	ntg@nic.surfnet.nl

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 tracht dat te bereiken door het uitwisselen van informatie, het organiseren van congressen, symposia en tentoonstellingen m.b.t. T_EX en 'T_EX-producten', en door het onderzoeken en vergelijken van T_EX met soortgelijke/aanverwante producten.

De NTG biedt haar leden ondermeer het volgende:

- Tweemaal per jaar een NTG-bijeenkomst.
- Tweemaal per jaar de uitgebreide NTG MAPS (Minutes and Appendices).
- Speciale MAPS uitgaven (o.a. T_EX cursusmateriaal en PR set).
- De 4allT_EX CD-ROM met een volledige en direct te gebruiken T_EX PC-DOS implementatie inclusief een zeer uitgebreide verzameling van utilities. De CD-ROM bevat ruim 100 Mbyte aan documentatie, inclusief discussielijsten van vele jaren, *alle* MAPS uitgaven, en zeer veel tutorials.
- De discussielijst TEX-NL waarop vragen gesteld worden. Ook worden er via deze listservers ervaringen uitgewisseld.
- De fileservers TEX-NL waarop algemeen te gebruiken 'T_EX-producten' staan. De meeste van deze T_EX-producten zijn, tegen geringe vergoeding, ook op diskette verkrijgbaar. Daaronder valt ook de 4allT_EX distributie set: een gebruiksvriendelijke en 'volledige' L^AT_EX/T_EX implementatie voor PC-DOS systemen.
- Het NTG FGBBS Bulletin Board met ruim 100 MByte aan T_EX en aanverwante software.
- Activiteiten in werkgroepen. Enkele belangrijke werkgroepen zijn: 'Nederlandse T_EX', 'PC's en T_EX', 'educatie' (cursussen), en 'communicatie'.
- Korting op (buitenlandse) T_EX congressen en cursussen, en op het lidmaatschap van TUG.
- Eenmaal per jaar een ledenlijst met per lid informatie welke software en welke hardware, in relatie met T_EX, wordt gebruikt.

Lid worden kan door overmaking aan de penningmeester van het verschuldigde contributie bedrag. Daarnaast dient een informatieformulier te worden ingevuld, welke laatste via het secretariaat te verkrijgen is.

De contributie voor een persoonlijk lidmaatschap bedraagt *f* 75,-, de contributie voor een instituutslidmaatschap bedraagt *f* 200,-. Een instituutslidmaatschap geeft het recht om drie personen aan te wijzen die informatie welke aan de leden wordt verstuurd, ontvangen. Van die drie personen dient één persoon te worden aangewezen als rechtsgeldige vertegenwoordiger van het bedrijf/instituut, een ander als vervangend vertegenwoordiger.

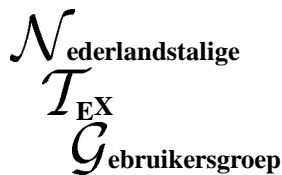
Indien meer leden per bedrijf/instituut lid willen worden, geldt als additioneel tarief *f* 50,- per persoon.

Voor studenten geldt eveneens een tarief van *f* 50,- (geen stemrecht; bewijs van inschrijving vereist). Voor afwijkende regelingen dient contact met het bestuur opgenomen te worden.

Een gecombineerd NTG/TUG lidmaatschap voor 1994 bedraagt *f* 171,- per jaar (i.p.v. *f* 75,- + \$ 60).

Belgische leden kunnen de lidmaatschapskosten van BF 1420 (individueel), BF 3790 (instituutslidmaatschap) of BF 3240 (NTG/TUG lidmaatschap) overmaken op de NTG Belgische postgiro te Veldegem (zie hierboven).

De statuten van de Nederlandstalige T_EX Gebruikersgroep zijn via het secretariaat of via de fileservers te verkrijgen.



- Aanwezig** : A. Al-Dhahir (UT); E. Algera (EGD); P. Bäumer; A.W.W.M. Biegstraaten (TUD); P. Bloemen (TUE); T. Bloo (Jonge Onderzoekers); E.P.M. Boets; P.J. Braakman; J. Braams (PTT Neher Lab.); R. Breedveld (TUD); L. de Coninck (de Kraal); N. Cox (KUN); M.M.M. Dings (AKZO Research); F. van Ditmarsch (EGD); W. Dol; W. Dolman (Hogeschool van Amsterdam); E. van Eynde (Univ. Rekencentrum); C.M. Fortuin (Hogeschool Gelderland); E. Frambach (RUG); M. van Geest (CA-WCS/Koninklijke Marine); W.L. Gilberts (Hogeschool van Amsterdam); F. Goddijn; W.J. van de Guchte; H. de Haan (TUD); J. Hagen (Pragma); S. van Harreveld (Actual Business Group); B.H. Hasselman; R. van der Heijden (Hogeschool Utrecht); J.B.W. Hoebeek (TUD); C.F.A. Janssen; R. de Jeu (Wolters Kluwer); A. de Jong (UU); W.J. Karman (KUN); G. Kiers; R. Klop (TUD); N.S. Kroonenberg (RUG); M.J. Krugers (Techn. Marketing Consulting); C.G. van der Laan; E.J.W. Lam; M. Lavaud (Univ. Orléans); A. de Leeuw van Weenen (RUL); H.A.N. van Maanen; H. van der Meer (UvA); A.J. de Meijer (UU); G.J.H. van Nes (ECN); A. van der Neut (ElectroGIG); J.H.B. Nijhof (RUG); G. Oomen (Wolters Kluwer); P. van Oostrum (UU); H. Op den Brouw (KUB); N.A.F.M. Poppelier(Elsevier); J. Pijnenburg (KUB); J. Renkema (Theol. Univ.); W. Rombouts (Océ); A. Schuitman; W. Smit; A. Soos (UT) J.H.M.J. Soutberg (ISYS Prepress Serv.); J.H. van de Stadt; G. Stappers; F. Terlouw (Océ); J.T.W.M. Tissen (RUG); J. Timmerman (O&W); P. Tutelaers (TUE); E. Ulijn (TUD); P. Vanoverbeke; M. van Veen; J.E. van Weerden (UU); E.A.G. Weits (NS); J.J. Winnink; W.J. Withagen (TUE); R. Zielman.
- Notulist** : Jos Winnink

1 Opening

Op deze tweede bijeenkomst in het lustrumjaar van de NTG, die gehouden wordt bij Océ in Den Bosch, zijn maar liefst 72 leden aanwezig.

Om 10:00 uur worden wij door de heer Rombouts van Océ welkom geheten. Alvorens de vergadering te openen stelt de voorzitter het bestuur voor aan de aanwezigen. In het openingsgedeelte wordt tijd ingeruimd voor een videopresentatie van Océ. Tevens deelt de voorzitter mee dat de redactie van de MAPS aan het einde van het openingsblok voor een kleine verrassing zal zorgen.

Hierna opent voorzitter Kees van der Laan de vergadering.

2 Verslag van de NTG bijeenkomst van 10 juni 1993

Bij het verslag van de NTG bijeenkomst van 10 juni 1993 zijn geen opmerkingen en aldus wordt het ongewijzigd goedgekeurd.

3 Ingekomen stukken en Mededelingen

De volgende mededelingen worden gedaan:

- Berichten van verhindering ontvangen van de volgende leden:
B. Bulder, B. Doppenberg, P. Eilers, E.G.M. Embsen,

J. van der Ende, E.J. Evers, A. Heijs, P.E.M. Huygen, T.A. Jurriens, W. Kraay, E. Luijten, W.J. Maas, R.D.A. Pauly, R. Smedinga, J. van der Zanden, R. Veldhuyzen van Zanten, B. Verhegge.

- Van een enkele buitenlandse zustervereniging (DANTE) zijn weer tijdschriften ontvangen. Deze liggen zoals gewoonlijk weer op de leestafel ter inzage. Ook diverse andere T_EX/L^AT_EX documenten kunnen ingezien worden
- Addison-Wesley is aanwezig met een uitgebreide boekentafel met ondermeer T_EX/L^AT_EX en PostScript boeken. Ook kan men reeds intekenen voor het L^AT_EX companion boek van Goossens, Mittelbach en Samarin. Boeken waren met 10% korting aan te schaffen.
- De secretaris merkt op dat hij de volgende zaken heeft meegenomen, die voor belangstellenden te koop zijn:
 - De originele versie van de syllabus van David Salomon voor de weggeefprijs van f 10,-. Er zijn nog slechts enkele exemplaren beschikbaar.
 - De volledig herziene versie van de syllabus van David Salomon voor de veeleisende T_EXer voor de prijs van f 50,-. Het is een boekwerk van meer dan 500 pagina's met hoogstaande T_EX-informatie.
 - De PR-set, waarvan alle leden inmiddels een exemplaar hebben ontvangen. Extra exemplaren zijn voor de prijs van f 2,50 te verkrijgen.

Editors van deze MAPS zijn: Wietse Dol, Gerard van Nes en Jos Winnink.

Oplage MAPS: 290.

Het verslag van de NTG bijeenkomst op 18 november 1993 is (in concept) eind maart 1993 via de post reeds gestuurd naar alle NTG leden.

- De voorzitter heeft nog de volgende mededelingen/verzoeken:
 - Het verzoek om *local guides* naar de redactie van de MAPS te sturen.
 - Dat vrijwilligers voor het verenigingswerk nog steeds welkom zijn.
 - In het voorjaar van 1994 zowel hijzelf als Jos Winnink zullen aftreden en niet herkiesbaar zijn. Het bestuur heeft voor de opvolging enige actie ondernomen en meldt dat:
 - Johannes Braams te kennen heeft gegeven de kandidatuur voor het voorzitterschap te willen aanvaarden.
 - Wietse Dol, die inmiddels ook in de redactie van de MAPS zit, kandidaat wil zijn voor het penningmeesterschap.
 - Frank Goddijn, wel bekend van het FGBBS, kandidaat wil zijn voor bestuurslid.

Het bestuur meende dat, hoewel de verkiezingen nog een half jaar weg zijn, het toch verstandig was in een vroegtijdig stadium naar kandidaten te zoeken. Dit neemt overigens niet weg dat ook andere kandidaten welkom zijn, mits tijdig aangemeld.

4 Begroting 1994

Wietse Dol merkt op dat bij de toelichting op de inkomsten een post staat van '3 aanvullingen' à *f* 50,- staat. Volgens hem moet dat *f* 150,- zijn en niet *f* 350,- zoals vermeld. Johannes Braams merkt op dat het om een drukfout gaat.

Op de vraag wat de post *sponsoring* omvat wordt geantwoord dat deze ondermeer een bijdrage aan het L^AT_EX3 project zal omvatten, als de vergadering daar later op de ochtend toe besluit.

Computerfaciliteiten. Om mogelijke problemen in de toekomst voor te zijn, is deze post als PM-post op de begroting gezet. Tot op heden is het gebruik van faciliteiten van SURF nog gratis.

De tweede PM-post (investeringen). Het bestuur merkt op dat de NTG nogal *hangt* op faciliteiten geboden door werkgevers van actieve (bestuurs)leden. Binnen het bestuur leeft dan ook de gedachte om eigen apparatuur aan te schaffen voor de vereniging om zo ondermeer de afhankelijkheid van werkgevers te verminderen. Als eerste zal hiertoe een laserprinter bij het NTG secretariaat geïnstalleerd worden.

Sacha van Harreveld vraagt zich af of we niet moeten streven naar een sluitende begroting. Johannes Braams merkt op dat dat op zichzelf wel zo is, maar dat de begroting met name aan de uitgaven kant zodanig ruim is dat er geen problemen verwacht worden. Ook is de financiële situatie zodanig dat we één à twee jaar de tijd hebben om de situatie aan te kijken. Op den duur zal een verhoging van de contributie vermoedelijk onvermijdelijk zijn.

Vervolgens wordt de begroting met algemene stemming goedgekeurd.

5 Wat verder ter tafel komt

Blijkbaar vond de vergadering dat er reeds voldoende op de tafel lag, want ze kwam niet met punten voor dit agenda punt naar voren.

6 L^AT_EX3

Jos Winnink doet verslag van de onlangs onder de leden gehouden enquête over het gebruik van T_EX/L^AT_EX. Een schriftelijk verslag wordt door hem toegezegd voor MAPS 94.1.

Vervolgens geeft Johannes Braams een korte uiteenzetting m.b.t. de situatie rondom het L^AT_EX3-project. De ontwikkelingsgroep is inmiddels twee keer bij elkaar geweest. Er is besloten om een interim release (L^AT_EX2_ε) uit te brengen. Een eerste publiekelijke testversie zal aan het eind van 1993 of in het begin van 1994 beschikbaar komen. Deze interim release heeft tot doel om te komen tot één *format*, zodat alle neuzen weer gericht zijn. Alleen NFSS2 wordt hierin standaard opgenomen. De stijlen 'array', 'multicol', 'theorem' en 'babel' zullen apart gedistribueerd worden. Ook is er overleg met Leslie Lamport en Addison Wesley geweest, omdat de ontwikkelgroep van mening is dat er ook een nieuwe versie van Lamport's boek moet komen.

De nieuwe versie van L^AT_EX zal een compatibiliteitsmode kennen zodat bestaande documenten nagenoeg ongewijzigd kunnen worden verwerkt. Hiertoe is de opdracht `\documentstyle` vervangen door `\documentclass` en `\usepackage`.

Discussies tussen de ontwikkelgroep en Leslie Lamport over wat er aan L^AT_EX moet worden toegevoegd en wat er moet worden gewijzigd, hebben geleid tot extra vertraging, maar de indruk is dat men het nu eens is.

In L^AT_EX3 komt ondersteuning voor *graphics*.

Er is veel tijd besteed aan het boek *The L^AT_EX companion*, van Goossens, Mittelbach en Samarin.

Ten gunste van het werk aan L^AT_EX2_ε is het werk aan L^AT_EX3 wat naar de achtergrond verschoven.

Een uitgebreid verslag van de activiteiten rondom het L^AT_EX3 project komt in TUGboat, TTN en in de MAPS. Er is duidelijk vooruitgang geboekt sinds de vorige NTG bijeenkomst.

Paul Braakman vraagt zich af of er afspraken komen voor de namen van *document classes* en *packages*. Johannes antwoordt hierop dat dit niet in de bedoeling ligt. Namen blijven vrij.

Phons Bloemen merkt op dat een mogelijke oplossing voor het naamgevingsprobleem gevonden zou kunnen worden door namen te laten claimen bij een *centrale instantie*.

Het voorstel van het bestuur om het L^AT_EX 3 project te steunen met een bijdrage van *f* 1000,- wordt met algemene stemmen aangenomen.

Johannes bedankt namens de projectgroep hiervoor de vergadering.

7 MAPS

Alvorens dieper op de verschillende werkgroepverslagen in te gaan, wordt eerst tijd besteed aan de verschillende onderdelen van de MAPS. Een klein en vluchtig lezersonderzoek wordt door de voorzitter gehouden onder de aanwezigen om na te gaan hoe de MAPS worden gelezen. Een eerste indruk:

MAPS 93.2 bijlage	Zwaartepunt onderwerp	Gelezen door (%)
A – D	(verenigingszaken)	100
E – J	(netwerkzaken)	60
K – P	(verslagen)	20
Q – Y	(\TeX / \LaTeX algemeen)	70
Z – \mathcal{E}	(PC-DOS implementatie)	85
\mathcal{F} – \mathcal{H}	(voor de gevorderden)	15
\mathcal{I} – \mathcal{L}	(idem; \TeX)	10
\mathcal{M} – \mathcal{Q}	(voor de gevorderden)	5
\mathcal{R} – \mathcal{U}	(boekbespreking; info)	20
\mathcal{V}	(NTG ledeninformatie)	10

De redactie van de MAPS bestaat op dit moment uit Wietse Dol, Gerard van Nes en Jos Winnink. Zij worden bij hun werk ondersteund door Frank Goddijn en Henk de Haan.

Inmiddels heeft ook de eerste redactievergadering plaatsgevonden en kan de *maps.sty* binnenkort worden vrijgegeven. De *gehele* MAPS 93.2 is met behulp van \LaTeX gepubliceerd, waartoe slechts een beperkt aantal artikelen behoefde te worden aangepast.

Om de omvang van de MAPS in de toekomst in de hand te kunnen houden en de kosten enigszins aanvaardbaar te maken, zal een deel van de informatie die nu met een zekere regelmaat in de MAPS wordt opgenomen, mogelijk in een aparte losse publikatie worden opgenomen. Te denken valt hierbij aan de ledeninformatie, de verschillende FAQ's e.d. Ook wordt geprobeerd om de artikelen zodanig te documenteren dat duidelijk is wat de beoogde doelgroep is voor het betreffende artikel. Ook zal getracht worden om een gerubriceerd overzicht te maken van reeds verschenen artikelen, die op verzoek, tegen kostprijs, kunnen worden verkregen.

Daarnaast heeft de redactie besloten om ter stimulering van auteurs over te gaan tot het instellen van een *MAPS-award*. Voor dit jaar zijn de genomineerden:

Piet van Oostrum, Robert Best, Kees van der Laan, Piet Tutelaers, Philippe Vanoverbeke, Phons Bloemen, Frank Goddijn, Henk de Haan, John Timmerman, Jules van Weerden, Johannes Braams, Wietse Dol en Rob de Jeu.

Het vele werk dat deze auteurs voor de MAPS in welke vorm dan ook, geleverd hebben, zorgde uiteindelijk voor een record aan pagina's vol met informatie voor zowel beginnende als gevorderde \TeX / \LaTeX gebruiker.

De jury (\equiv MAPS-redactie) heeft na enige discussie gemeend om de eerste MAPS Author Award voor 1993 (een grote zak spekkies; symbool voor het gehele \TeX gebeuren) toe te kennen aan student Phons Bloemen, wegens zijn inzet als betrekkelijk jong lid bij het katalyseren van

een reeds langlopende activiteit waar een zeer grote groep van (DOS gebruikende) NTG-ers al een hele tijd op zat te wachten.

Jules van Weerden merkt op dat Kees van der Laan op de NLUUG-dag een award heeft gekregen voor *het beste artikel* uit de NLUUG-proceedings. De aanwezigen reageren hierop met een ovationeel applaus.

Het verzoek van de redactie is om de *maps.sty* te gaan gebruiken voor artikelen, zodra deze beschikbaar is.

Met betrekking tot de MAPS is nog vermeldenswaard dat in het verleden vaak eerder verschenen artikelen werden overgenomen. Ook vindt vaak voorpublicatie plaats in de MAPS. Denk aan het artikel van Taupin over Music \TeX dat onlangs in TUGboat verscheen en reeds 2 jaar geleden in de MAPS stond.

8 Verslag bijeenkomsten

8.1 TUG

Op de van 26–30 juli 1993 gehouden TUG'93 bijeenkomst te Aston (zie bijlage N van MAPS 93.2) merkte Malcolm Clark op dat de NTG de best geïnformeerde \TeX -gebruikersgroep is.

8.2 CyrTUG

Over de CyrTUG bijeenkomst van 4–7 oktober 1993, bijgewoond door Kees van der Laan (zie ook bijlage P in MAPS 93.2), werden geen vragen gesteld.

8.3 NLUUG

Een uitgebreid verslag van deze in november 1993 gehouden bijeenkomst te Ede van de Nederlandse UNIX gebruikersbijeenkomst (lezingen door Kees van der Laan en Piet van Oostrum; postersessie door het \LaTeX team) zal in de volgende MAPS verschijnen.

De NLUUG telt ongeveer 500 leden; op de NLUUG-dag waren ook 500 deelnemers aanwezig. De Proceedings liggen op de leestafel ter inzage. De speciaal voor deze dag vervaardigde CD-ROM is ook nog na te bestellen voor f 35,-. Mogelijk dat de proceedings ook te bestellen zijn, maar de NLUUG moest hier nog over nadenken.

9 Verslag werkgroepen

9.1 PC-zaken

4all \TeX bevindt zich in een β -test en omvat nu 28 diskettes van 1.44 Mbyte. Het ligt in de bedoeling om twee keer per jaar een update uit te brengen omdat ook leden die niet op een netwerk zijn aangesloten behoefte hebben aan up-to-date informatie.

Frans Goddijn (FGBBS) krijgt ter vergadering het eerste exemplaar onder luid applaus overhandigd. Het tweede exemplaar is voor de secretaris. Binnen enkele dagen zal 4all \TeX ook op de ftp-sites beschikbaar zijn. Omstreeks de jaarwisseling gaat er een versie richting Oost Europa ter ondersteuning van de aldaar plaatsvindende \TeX -activiteiten. De vernieuwde documentatie zal naar verwachting ook begin van het komende jaar beschikbaar zijn.

Het ligt in de bedoeling om het $\Delta\text{T}\text{E}\text{X}$ -manual als MAPS-*special* uit te brengen.

Voorts wordt het verzoek gedaan om $4\text{allT}\text{E}\text{X}$ ondersteuning voor met name: MS-Windows, OS/2 en LINUX. Gevraagd wordt naar vrijwilligers hiervoor.

Hans van der Meer brengt onder de aandacht dat $\text{OzT}\text{E}\text{X}$ (voor de Apple Macintosh) vernieuwd en uitgebreid is. De huidige versie heeft versienummer 1.6. Hij heeft een update van $\text{OzT}\text{E}\text{X}$ (één diskette) die hij voor verdere distributie aan de secretaris overhandigd.

Uit de vergadering komt de opmerking dat er ook voor andere platforms dan MS-DOS behoefte is aan ondersteuning. Te denken valt met name aan de Macintosh en de Atari. Opgemerkt wordt dat $4\text{allT}\text{E}\text{X}$ voor een groot gedeelte niet specifiek voor PC's is en derhalve ook op andere platforms bruikbaar is.

Alaaddin Al-Dhahir doet het verzoek om op het net ook vriendelijk te zijn voor PC-gebruikers, er waren niet lang geleden enkele minder prettige discussies op het net.

9.2 Communicatie: FGBBS

Frans Goddijn geeft het woord aan Henk de Haan. De zojuist ontvangen $4\text{allT}\text{E}\text{X}$ zal zo spoedig mogelijk (binnen enkele dagen) op FGBBS beschikbaar komen. $\text{OzT}\text{E}\text{X}$ 1.6 is inmiddels ook beschikbaar.

Voor NTG-leden is toegang tot het *internet* gerealiseerd en ook COMP.TEXT.TEXT is beschikbaar. Inmiddels geeft het FGBBS ook de mogelijkheid om diskettes te bestellen.

Johannes Braams informeert naar het bijhouden van het FGBBS. Henk de Haan antwoordt hierop dat hij files uit de CTAN servers haalt en regelmatig controleert of er op deze servers nieuwe bestanden staan dan wel dingen zijn gewijzigd.

Piet van Oostrum merkt op dat er eigenlijk een programma zou moeten zijn zodat je automatisch zou kunnen bepalen of je over de meest recente versie van een pakket beschikt.

Opgemerkt wordt dat CTAN gebruik maakt van *mirroring* en dat er een *mailing list* is waarop melding wordt gemaakt van *updates*.

Het aantal BBS gebruikers schommelt gewoonlijk rond de 15 per week.

9.3 Communicatie: TEX-NL, CTAN

Sinds vorig jaar (Portland) is CTAN na jaren van discussie beschikbaar gesteld. Dit betekent dat de drie grote archieven met TEX informatie door middel van automatische updates worden gesynchroniseerd.

Henk de Haan vraagt naar de mogelijkheid om het CTAN archief op CD-ROM te verkrijgen. Mogelijk komt TUG met een dergelijke CD-ROM.

Van de aanwezigen heeft zo'n 40 à 50 de beschikking over een CD-ROM drive (dan wel binnen een half jaar). Een dertigtal maakt regelmatig gebruik van CTAN. Een twintigtal aanwezigen gebruikt regelmatig het ftp-archief in Utrecht. Opgemerkt wordt dat *BITNET adressen* bij voor-

keur zo snel als mogelijk veranderd moeten worden omdat mogelijk binnen één à twee jaar BITNET in elk geval in Nederland zal verdwijnen (eventueel hiervoor contact opnemen met Jules van Weerden).

9.4 Nederlandstalige TEX

De situatie met betrekking tot de afbreekpatronen is wat sluimerend meldt Piet Tutelaers. Het probleem is de ondersteuning van de DC-fonts. De DC-fonts maken nog geen onderdeel uit van *standaard* TEX . Mogelijk kan door het werk dat in Duitsland wordt gedaan aan de DC-fonts binnenkort wel collectief worden overgegaan. Ook is de situatie m.b.t. de tekenverzameling ISO-latijn nog onduidelijk.

Piet Tutelaers heeft de uitgever van het 'Groene Boekje' benaderd met de vraag of de lijst met afbreekpatronen beschikbaar was. Het antwoord was ontkenkend. Mogelijk moet de vraag langs een officiële weg worden gesteld, bijvoorbeeld door het bestuur van de NTG.

Overigens is het voordeel van een *public domain* lijst dat je deze zelf kunt onderhouden.

10 Rondvraag

Nico Poppelier had iets willen meedelen over recente ontwikkelingen bij TUG. Maar omdat dit agenda punt naar het eind van de middag was verplaatst, en hij helaas niet meer aanwezig was om zijn zegje te doen, verviel dit punt.

Kees van der Laan kan met betrekking tot TUG nog het volgende meedelen:

- Volgend jaar bestaat TUG 15 jaar. Dit zal worden gevierd in Santa Barbara.
- TUGboat en TTN blijven beide in regelmaat verschijnen.
- Er wordt gewerkt aan een PC-versie van TEX .
- Een CD-ROM distributie is in de maak. Gedacht wordt aan een prijs van \$ 100.
- Een aantal 'getrouwe' werkers, waaronder Alan Hoenig, heeft zich teruggetrokken.
- De verspreiding van de TUGboat in Oost Europa heeft de aandacht.

John Timmerman stelt de vraag of er mensen zijn die ervaring hebben met het in een WP-bolwerk promoten van $\text{T}\text{E}\text{X}/\text{L}\text{A}\text{T}\text{E}\text{X}$.

Kees van der Laan antwoordt hierop:

- Er moet nog steeds een document komen met als titel: 'Waarom TEX '.
- Langzamerhand komt er steeds meer informatie op de plank te liggen die voor dit soort zaken gebruikt kan worden.
- De NTG kan docenten leveren.

Daarnaast wordt opgemerkt dat je het niet op cursussen of training moet gooien. Je moet de voordelen proberen te benadrukken, zoals het verwerken van 'oude' documenten. Er zijn diverse hulpmiddelen beschikbaar. Zo is er een $\text{WP2L}\text{A}\text{T}\text{E}\text{X}$ convertor en zijn er *frontends* voor $\text{T}\text{E}\text{X}/\text{L}\text{A}\text{T}\text{E}\text{X}$ beschikbaar.

Piet Tutelaers merkt op dat er op de NLUUG dag door Johan Vromans een *frontend* werd gedemonstreerd met de naam ‘Squirrel DAB’, dat in staat is om een ASCII-tekst naar L^AT_EX om te zetten. Het programma zou in het *Public Domain* komen.

Willem Smit vindt dat je moet beginnen met L^AT_EX. Gerard van Nes stelt dat je het (= L^AT_EX) resultaat bij printers moet laten rondslingeren, ‘dan verspreidt het zich vanzelf als een virus’.

Voorzover bekend is er slechts één organisatie (CWI) die de overgang naar T_EX/L^AT_EX centraal heeft doorgevoerd.

Discussies met het management worden in het algemeen niet als zinvol ervaren.

Dit soort discussies gelieve men zeker op TEX-NL te voeren, omdat dan meer mensen er wat aan hebben.

11 Voordrachten:

Op het programma staan de volgende voordrachten:

- ‘4T_EX: a T_EX workbench for MS-DOS PC’s’, door Wietse Dol aan de toetsen en Erik Frambach voor de microfoon.
- ‘Scientific Working Environment: A^ST_EX’, door Michel Lavaud.

De 4T_EX presentatie en demonstratie blijkt een zeer groot succes te zijn. De belangstelling voor de software was ook zeer groot. A^ST_EX daarentegen kent de nodige beperkingen en zal daarom ook zeker geen grote markt binnen Nederland hebben.

12 Volgende bijeenkomsten

De voorlopige planning van de volgende bijeenkomsten ziet er als volgt uit:

- **voorjaar 1994**
Rijksuniversiteit te Groningen op 9 juni 1994.
Onderwerp: T_EX/L^AT_EX-, METAFONT- en gereedschappen-educatie.
- **najaar 1994**
Universitaire Instelling Antwerpen (UIA) op 17 november 1994.
Onderwerp: ‘T_EX en publiceren’.
- **voorjaar 1995**
Universiteit Twente op 24 mei 1995.
Onderwerp ‘Multimedia en T_EX’ (onder voorbehoud).

13 Sluiting

Om ongeveer 17:00 uur wordt de dag door de voorzitter, na Océ bedankt te hebben en de deelnemers een goede reis toegewenst te hebben, gesloten.

Hierna gaat een gedeelte van de deelnemers nog natafelen en van een ‘Beaujolais Primeur’ genieten.

De volgende vergadering is op:

donderdag 9 juni 1994

te Groningen. Het thema: ‘T_EX en Educatie’.

1 T_EX kalender 1993/1994

31/4	jul/aug '94	TUG'94	Santa Barbara CA, USA
26/30	sep '94	EuroT _E X'94	Polen
17	nov '94	NTG (14 ^e)	Antwerpen
24	mei '95	NTG (15 ^e)	Twente

2 Glossary

Gebruikersgroepen

TUG	:	T _E X Users Group
LUG	:	Local Users Group
CSTUG	:	LUG Tsjecho Slowakije
CyrTUG	:	LUG USSR (het Cyrillisch taalgebied)
DANTE	:	LUG Duitsland (het Duits taalgebied)
GUTenberg	:	LUG Frankrijk (het Frans taalgebied)
HunTUG	:	LUG Hongarije
ITALIC	:	LUG Ierland
JTUG	:	LUG Japan
Nordic	:	LUG Scandinavië, Denemarken, en IJsland
NTG	:	LUG Nederland en België
SibTUG	:	LUG Siberië
UKTUG	:	LUG Engeland
YUNUS	:	LUG Turkije (feitelijk een discussielijst)
GUST	:	LUG Polen

Bulletins/journals

Baskerville	:	UKTUG
Cahiers GUTenberg	:	GUTenberg
Zpravodaj	:	CSTUG
TeXnische Komödie	:	DANTE
TeXline	:	Malcolm Clark; UK
GUST bulletin	:	GUST
TTN	:	T _E X and TUG News; TUG
TUGboat	:	TUG
MAPS	:	Minutes and Appendices; NTG

Diversen

AMS	:	American Mathematical Society
BoD	:	Board of Directors
SGML	:	Standard Generalized Markup Language
ltxiii	:	L ^A T _E X 3.0
FGBBS	:	NTG's Bulletin Board

3 NTG's T_EX Bulletin Board System

Op het T_EX Bulletin Board van de Nederlandstalige T_EX Gebruikersgroep (FGBBS) is een zo volledig en actueel mogelijke T_EX, emT_EX, L^AT_EX, TEX-NL en MusicT_EX collectie beschikbaar voor alle bezitters van een modem. Het BBS is kosteloos toegankelijk voor iedereen en er zijn geen beperkingen aan de hoeveelheid bestanden die kunnen worden opgevraagd. Het systeem is aangesloten op een High Speed modem, vergeleken met de transmissiesnelheid die een directe Internet link biedt misschien niet geweldig, maar veel beter kan het niet over de gewone huis-tuin- en keukenPTTlijn. De beheerders zijn Frans Goddijn en Henk de Haan.

FGBBS is te bellen op 085-217041.

Zie ook *Bijlage H*.

4 T_EX/L^AT_EX en Silicon Graphics

Van NTG lid Anton Heijs ontvingen wij het volgende bericht.

De Nederlandse Silicon Graphics Gebruikers Groep (SG³) is opgericht op 28 februari 1991.

De beoogde doelstellingen zijn die zoals vermeld in de uitnodiging van de oprichtingsbijeenkomst en kunnen als volgt worden geresumeerd:

- delen in gemeenschappelijke kennis en *legale* software (sourcecode en documentatie),
- organiseren van onderlinge contacten middels 4 jaarlijkse evenementen en e-mail communicatie,
- belangen van Silicon Graphics gebruikers behartigen,
- verzorgen van een nieuwsbulletin,
- en vele andere activiteiten (zie b.v. schrijven van 12 februari 1991).

De meetings, waarvan het bestuur er minimaal 2 per jaar wil organiseren, zullen 1 dag duren, van 10:00 tot 16:00 uur. Er wordt gedacht aan lezingen van experts, presentatie van demo's, aandacht voor gebruikersproblemen en hun oplossingen en natuurlijk het ontmoeten van andere Silicon Graphics gebruikers.

De nieuwsbrief maak ik altijd met L^AT_EX een versie die ik van Piet Tutelears heb gekregen. Binnen de SGI gebruikers bleek er toen ook vraag te zijn naar een tape met T_EX versie 3.141 voor SGI en zodoende heb ik een aantal tapes gemaakt om rond te zenden, zoals net aan uw MAPS-editor, Gerard van Nes. Omdat een gebruikers gemeenschap (SGGG en NTG) het moet hebben van het delen van gezamenlijke kennis en tools, wil ik bij deze hier melden dat alle geïntereseerde bij mij kunnen aankloppen voor deze tape of voor SGI aangelegenheden (ftp-sites, pd-tools en natuurlijk lidmaatschap).

Anton Heijs (editor SGGG newsletter), Staring Centrum, afdeling Micromorfologie & Bodemstructuur, Marijkeweg 11, 6700 AC Wageningen, tel: 08370-74596, e-mail: a.w.j.heijs@sc.agro.nl

of anders:

SG³ p/a T.U.D., faculteit der Technische Natuurkunde,

vakgroep Computational Physics, Lorentzweg 1, 2628 CJ Delft, t.a.v. A.F. Bakker (voorzitter van de SGGG).

5 NTG's winkel

Via de NTG is beschikbaar:

- **4all \TeX CD-ROM:**

Ruim 470 Mbyte (ongeveer 19.000 files) aan ondermeer \TeX utilities, fonts, \TeX / \LaTeX /METAFONT/etc documentatie, de volledige MAPS 1 t/m 12, discussielijsten \TeX -NL, \TeX -HAX, UK \TeX van de afgelopen 5 jaren, etcetc.

Kosten inclusief het uitgebreide \TeX manual: f 60,-.

- **Syllabus Advanced \TeX course:**

Insights and Hindsights, David Salomon (*revised*; ruim 500 pagina's).

MAPS'92 speciale uitgave.

Kosten f 50,- voor leden en f 60,- voor niet-leden (extra verzendkosten: f 10,-).

- **PR set MAPS'93 speciale uitgave:**

Ruim 25 pagina's; 1 exemplaar gratis voor leden; extra exemplaren: f 2.50; niet-leden: f 5,- (extra verzendkosten: f 5,-).

Bestellingen kunnen gedaan worden door overmaking van het verschuldigd bedrag (plus verzendkosten) op de postgiro van NTG (1306238) t.n.v. de penningmeester Johannes Braams te Zoetermeer, met vermelding van hetgeen gewenst is.

6 4all \TeX : NTG's \TeX voor MS-DOS

In de vorige MAPS (93.2) is informatie te vinden over de \TeX MS-DOS implementatie 4all \TeX . De volledige distributie set van 30 high-density (1.44 Mb) diskettes is op te vragen bij de secretaris van de NTG. Daarnaast is de complete 4all \TeX distributie set te vinden op het FGBBS (085-217041).

7 Oproep Local Guides

Regelmatig blijkt dat op verschillende plaatsen erg goede 'Local Guides' beschikbaar zijn. Helaas heeft (voor zover bekend) niemand een overzicht van wat er allemaal op de markt is. Daarom wil de NTG een inventarisatie maken, die te zijner tijd gepubliceerd zal worden.

Graag zou de NTG de volgende gegevens ontvangen van alle in omloop zijnde Local Guides:

titel; auteur(s); doelgroep (beginners/gevorderden, \TeX , \LaTeX , Metafont, etc.); (ruwe) inhoud; taal (Nederlands, Engels, etc.); aantal blz.; prijs; hoe te bestellen; contactpersoon;

Stuur de gegevens naar:

Erik Frambach
e.h.m.frambach@eco.rug.nl
Rijksuniversiteit Groningen,
vakgroep Econometrie,
postbus 800, 9700 AV Groningen
tel. 050-633785

8 NTG/TUG lidmaatschap

Het blijkt soms dat nieuwe NTG/TUG leden na ongeveer een half jaar nog geen TUGboat of TTN van TUG hebben ontvangen. Ondanks dat men een TUG lidmaatschap via NTG aanvraagt, blijkt in bijna alle gevallen de administratie- en verzendproblemen bij TUG zelf te liggen. Mocht na enige tijd geen post van TUG ontvangen worden, dan worden de betreffende NTG/TUG leden dringend verzocht om contact op te nemen met de penningmeester van de NTG: Johannes Braams.

9 MAPS 94.2

Sluitingsdatum voor het inleveren van artikelen, bijlagen, en/of mededelingen voor de volgende MAPS uitgaven is: 1 oktober '94 (MAPS 94.2) en 1 april '95 (MAPS 95.1).

Aanleveren kopij voor de komende MAPS:

- Bij voorkeur in \LaTeX gebruikmakend van de: `maps.sty`
Deze stijlfile is via de redactie te verkrijgen en beschikbaar op de \TeX -NL fileservers, archive.cs.ruu.nl (ftp-site) en FGBBS (085-217041).
Daarnaast kunnen bijdragen ingestuurd worden gemaakt met `ltugboat.sty` of `article.sty/report.sty`.
- Verder zijn bijdragen vanzelfsprekend ook welkom in \TeX of ongeformatteerd.
- Plaatjes bij voorkeur als (Encapsulated) PostScript file.

Bij onduidelijkheid gaarne contact opnemen met de redactie.

Daar MAPS bijdragen in *plain* \TeX worden omgezet in \LaTeX , verdient vanzelfsprekend aanbieding van materiaal in \LaTeX de voorkeur!

Eventuele nadere richtlijnen voor auteurs zijn op te vragen bij de redactie.

Bijdrage kunnen gestuurd worden naar:
vannes@ecn.nl

Niet Internet-gebruikers kunnen hun bijdrage ook via modem/PTT lijn direct naar de redactie sturen. Gaarne hiervoor eerst contact opnemen met Gerard van Nes.

Van de Voorzitter. . .

Activiteiten

NTG's 4all \TeX CD-ROM gaat het helemaal maken, daar ben ik van overtuigd. Afgelopen November hebben wij kennis genomen van de 4all \TeX functionaliteit, levendig voorgetoverd door Erik en Wietse.

Komende Juni, om precies te zijn de 10de, kunnen wij onder begeleiding hands-on ervaring ermee op doen.

Het tastbare resultaat van de cursus is in ieder geval het CD-ROMmetje in de tas en de documentatie onder de arm, onder het bewustzijn dat het (\LaTeX) \TeX -en met deze werk-omgeving stukken eenvoudiger is geworden. Later in het week-end zal men ongetwijfeld op de CD-ROM rondzwerfen, voluit en voor de wind, op zoek naar ander fraais, je weet maar nooit.

Moeder de vrouw, en andere huisgenoten, zullen zich zo langzamerhand toch wel afvragen wat voor een magisch persoon die Don Knuth wel niet moet zijn, die een \TeX ie zo lang aan het scherm gekluisterd kan houden, ongevoelig voor elk signaal uit de directe omgeving, als een kat loerend op zijn prooi.

Ach wij moeten nu maar eens met SURF gaan babbelen en dit fraais niet alleen voor onszelf houden.

De CD-ROM wordt voor de Europese markt op de Euro \TeX '94 gelanceerd. Erik en Wietse hebben inmiddels toegestemd daar als 'invited speaker' op te treden. Hulde heren, ere wie ere toekomt! Deze lof geldt ook voor Phons natuurlijk, en ook Maarten zijn wij dank verschuldigd, al blijven zij wat op de achtergrond.

MAPS.sty.

Lang gewacht en stil gezweven—nou ja—hij is er dan toch maar onze eigen MAPS.sty, voor \LaTeX . Dank Gerard en Wietse! Persoonlijk, zou ik de inner formatting zaken onafhankelijk van \LaTeX willen hebben. Bijvoorbeeld voor verbatim niet \LaTeX gebonden macros gebruiken, maar macros in plain, met de eis dat ze in elke omgeving gebruikt kunnen worden. Verder heb ik natuurlijk nog wel wensen, c.q. ideeën, maar dat moeten wij maar eens in de Evenaar doorpraten op 9 Juni bijvoorbeeld.

Komende bijeenkomsten.

Op 9 Juni is de NTG voorjaarsbijeenkomst met als thema Educatie. Piet van Oostrum zal ons op de hoogte brengen van de perikelen rondom de installatie van \TeX vooral op UNIX systemen. Erik-Jan Vens zal ons bijpraten over de ontwikkelingen op het gebied van fonts. Johannes Braams zal eens haarfijn het hoe en waarom van document classes uit de doeken doen. En ondergetekende zal de vergeten bladzijde Manmac in het juiste daglicht plaatsen. De vol-

gende dag is er de low-budget 4all \TeX cursus. Kortom een interessant educatief gebeuren al met al.

In November zullen wij bij onze zuiderburen op bezoek gaan—Universitair Instituut Antwerpen om precies te zijn—met als thema Publiceren met (\LaTeX) \TeX . Volgend voorjaar zijn wij te gast bij de Technische Universiteit Twente. Het thema van die bijeenkomst luidt Multi-media en \TeX . Suggesties voor bijdragen zijn welkom.

4all \TeX .

Na de laatste bijeenkomst, en mogelijk door het voorbeeld van de NLUUG CD-ROM gestimuleerd, is het idee van een NTG CD-ROM geboren. Kinch had in Aston al een 'TeX' CD-ROM, en het CTAN wordt op CD-ROM gezet. Het gonst alom dat dit nieuwe medium uitermate geschikt is als extern geheugen bij een PC. Zeg maar zo'n 600MByte, met redelijk snelle toegang, gepaard aan compactheid, redelijke prijs en lange levensduur. Wat wil een mens nog meer?

Ook TUG is bezig op dit vlak—lees geeft NTG het voortouw—om te komen tot een TUG CD-ROM set, de canonieke \TeX and Metafont set. Hier liggen de zaken echter wat ingewikkelder, omdat men het (nog) grootschaliger wil aanpakken. Niet alleen (\LaTeX) \TeX etc. voor een DOS omgeving maar voor vele platforms. En dit alles zo uniform mogelijk. Kortom een haast onmogelijke opgave. Dan is er nog het selectie probleem, onder de beperking dat een aantal goede omgevingen niet meer vrij zijn. Enerzijds is er veel en anderzijds moet het goede gekocht worden. Over goede documentatie en het geheel op bruikbaarheid te verifiëren zal ik verder maar zwijgen. Inmiddels is er een CD-ROM discussielijst geopend door George Greenwade. De discussie hierop lijkt erop uitgelopen te zijn dat niemand meer weet wat de meest geschikte directory structuur is voor zo'n brede set. Het blijft moeilijk constructief en effectief via een discussielijst de gang erin te houden.

Aan de andere kant moest de markt afgetast worden. Welke LUGs willen participeren en in welke mate? Wat zou men er graag op willen hebben, en voor hoeveel exemplaren zou men willen intekenen? Om maar een paar harde vragen te noemen. Christina Thiele heeft daartoe een fraaie email/brief uitgezonden, waarvan ik de response niet weet. Informeel weet ik dat UKTUG—vooral de duizendpoot Sebastian Rahtz—de Nordic Group—ons aller Roswitha Graham—en GUTenberg—o.a. in de persoon van Michiel Lavaud—geïnteresseerd zijn. De positie van de Oost- en Midden-europeanen wordt nog wat bemoeilijkt doordat de CD-ROM driver als algemeen goed nog wat ver af is, nog verder dan bij ons. Parallel speelt dat de NTS groep zich ook gebogen had over de TUG (\LaTeX) \TeX set, en natuurlijk

ook op de CD-ROM als medium is uitgekomen. In hoeverre TUG, NTS en de CD-ROM-ers op een lijn zitten, weet ik eerlijk gezegd niet. In het begin doemde er wel iets van competentie c.q. competitie op.

De consequentie is dat het wat meer tijd vraagt om tot een realistische doelstelling te komen, dan wel zoveel mogelijk betrokkenen op een lijn te krijgen (en te houden).

Het leek ons nuttig ondertussen wat ervaring op te doen met de beperktere NTG 4all \TeX CD-ROM. Gewoon praktisch bezig zijn en de ervaring t.z.t. inbrengen kan mijns inziens nooit kwaad. Sterker nog dat is toch voetje-voor-voetje voortgaan in de goede richting. De logistiek van 4all \TeX alleen al wordt veel handzamer via CD-ROM en kan economisch uit. Daarnaast bevat het extras, zoals: het \TeX book en het Metafont book, veel nuttigs uit het wel en wee van de NTG, en andere goodies. Een waar verzamelingsobject.

NTG-ers go for it!

PR set, het kan verkeren.

Wat dacht je wat? De PR set is in elektronische vorm beschikbaar, op de CD-ROM natuurlijk. 600dpi voor de gedrukte vorm en in verband met gangbare printers 300dpi voor de CD-ROM (METAFONT code aanpasbaar natuurlijk). Dank Jos (en auteurs) voor al het werk! Leden die een bijdrage hebben worden als immer aangemoedigd contact op te nemen met de redactie van de MAPS, en niet te wachten tot de volgende bijzondere gebeurtenis.

TUG

De grenzen worden in Santa Barbara, TUG '94, wederom verlegd. \TeX in geuren en kleuren!

Voor 1995 is de annual meeting in Florida, en daarna in Europa. Het komt dus steeds dichterbij. Ondertussen zijn nuttige documenten ontwikkeld, de zogenaamde Info-sheets, met informatie over wat er op (L \AA) \TeX gebied beschikbaar is voor de diverse systemen. Ongetwijfeld zal dit t.z.t. opgenomen worden in de resource directory.

Wat mij treurig stemt is dat er nog steeds niets geregeld is met CyrTUG over de beschikbaarheid in elektronische vorm van de TUGboat inhoud. Zelfs geen gedeeltelijke oplossing om wederzijds de goede wil te tonen, en te kijken hoe het loopt in de praktijk.

tug.ppt proposal.

Zelf heb ik mijn studie over de TUGboat stijlen—en afgeleiden—aangeboden aan de BoD van TUG met het minimale verzoek er kennis van te nemen, en een TUG preprint stijl, zeg tug.ppt, te adopteren. Het laatste is nuttig omdat er voor auteurs dan geen verschil meer is in tugboat.sty en tugproc.sty, om maar iets concreets te noemen. Mijn dagdroom is dat LUGs deze preprint stijl idee in plaats van, c.q. naast, hun stijl adopteren, en daarmee tonen compatibel met elkaar te willen zijn. Een auteur kan dan zijn copy met ongewijzigde markup aan iedere LUG/TUG aan-

bieden. Meestal zijn dromen bedrog. Wij zullen zien. Ik hoop in ieder geval dat NTG's redactie het idee adopteert, evenals TUG.

Prototype, revise, rewrite.

Daarnaast doe ik in de studie nog veel meer suggesties, zoals de *verbatim mode*, om onafhankelijk te zijn van of je nu in \TeX , L \AA \TeX , of ... \TeX werkt.¹

De rode draad door mijn conclusies is het geheel eenvoudiger te maken. Mijn tug.ppt voorstel is ruwweg 100 regels, in ieder geval een educatief opstapje. Zelf gebruik ik dit voor mijn verhalen nu.

LUGs

UKTUG is geweldig actief. De Baskerville verschijnt weer regelmatig met een fraaie inhoud en dito opmaak. Hulde. Ook hebben zij een indrukwekkende lijst van bijeenkomsten met boeiende themas. De primeur van de L \AA \TeX 2 ϵ cursus is algemeen bekend. Dat daarnaast Sebastian BoD lid is en betrokken bij de TUG '94, de CD-ROM, CTAN e.d. activiteiten, doet je alleen maar afvragen hoe hij dat allemaal klaar speelt. En dan zwijg ik nog maar over de prima samenwerking die hij weet te onderhouden, o.a. met onze redactie. Gelukwensen!

Ook heb ik gehoord dat GUTenberg de spits zal afbijten met de Cahiers in kleur! Ik ben benieuwd.

DANTE's Stammtisch blijft trekken, alleen is het er weer niet van gekomen om eens aan te schuiven.

GUST laat steeds meer van zich horen. CyrTUG krijgt al onderafdelingen, alhoewel hun ledental vergelijkbaar met ons is.

Euro \TeX '94

Het verheugende nieuws is dat GUST—lees Włodek Bzyl—de Europese September 1994 bijeenkomst op een Baltisch waddeneiland organiseert. Nog verheugender voor NTG is dat Erik en Wietse met hun 4all \TeX geïnviteerde sprekers zijn. En, als het even meezit zullen zij er de Groningse cursus herhalen, low-budget vanzelfsprekend.

Mijn persoontje zal de TUGboat BLUES kwinkeleren, en vooral de idee van de xxx.ppt stijl uitbazuin. Ook heb ik aangeboden een Manmac workshop te houden.

Future of \TeX

Er is van alles gaande L \AA \TeX 2 ϵ , NTS, PostScript \leftrightarrow \TeX (kleuren!), ontpoppende LUGs en dergelijke. Wat er naar mijn mening over het hoofd wordt gezien is dat Knuth al lang voor een toekomst van \TeX gezorgd heeft, door het te koppelen als formatter aan zijn WEB systeem voor literate programming. Het formatteren van een bepaalde klasse van documenten heeft hij ondergeschikt gemaakt aan een groter doel: fraai gedocumenteerde software structureel

¹ Dit overigens in navolging van Knuth's latere werk, o.a. de markup van het CWEB manual.

makkelijker te maken en te onderhouden, better known als literate programming.

GNA—global network academy.

Binnen de LP discussielijst is een interessant idee geboren, een nieuwe mogelijkheid van het gebruik van het internet: de GNA—global network academy—voor de educatie m.b.t. literate programming. Marcus Speh is de geestelijk vader en leider.

De wereld om ons heen

Door de oogharen heenziend zijn wij helemaal bij met (L^A)T_EX, etc. Het netwerk en de BBS zijn niet meer weg te denken. NTG is welbekend in Oost en West. Naar de MAPS wordt uitgekeken. 4allT_EX gaat op zwerftocht door Europa en veroverd de wereld, zodat wij ons weer kunnen richten op wat wij er eigenlijk mee willen doen.

Niettemin gaf de WEPT folder—a Week of Electronic Publishing and Typography—stof tot overpeinzing. Qua titel en inhoud zou dat de belangrijkste bijeenkomst op het computer-ondersteund documentgebeuren moeten zijn. Is dat ook zo? Hoe zit dat eigenlijk?

Het is 4-in-1. RIDT94 legt de nadruk op ‘rendering’ en font kwaliteit. PODP94 is een workshop over ‘. . . document formatting, conversion, representation, recognition, en retrieval’ gerelateerd aan hypertext. TEP94 kondigt aan ‘The teaching of multimedia is focussed upon as a form of electronic publishing that presents its own challenges.’ Dan is er nog de EP94 met als trekker ‘. . . extended its range of topics to include hypertext and hypermedia systems, document recognition and analysis, and application of database techniques to document handling.’

Fraai, prachtig, daar is de research kennelijk op gericht. Hypermedia en hoe om te gaan met de complexiteit, hoe te navigeren op de woelige baren van het vele. (L^A)T_EX en Metafont als zodanig komen nauwelijks voor noch literate programming.² Is dit wel een getrouwe afspiegeling, of heeft men toch geen weet van wat er werkelijk gaande is? Ik bedoel Adobe was de trendsetter de afgelopen decade, en waar is hun recente Portable Document Format?

Voor de doorsnee wetenschapper, om maar een belangrijke klasse van (L^A)T_EX gebruikers te noemen, zijn (L^A)T_EX en Metafont de gereedschappen, de fixed points, waar men jaren mee vooruit kan, in samenwerking met andere goodies zoals PostScript, PDF, en de hypernavigatie in een World-Wide-Web.

Conclusie.

De WEPT is voor de happy few, en niet voor de zwijgende meerderheid die dagelijks resultaten met (L^A)T_EX uitwisselen via het internet, bijvoorbeeld.

Meetinstrumenten en de invloed op de meting

Dit verschijnsel is algemeen bekend binnen de experimentele natuurkunde. Wat heeft dit nu met (L^A)T_EX te maken? Ik moest hieraan denken toen ik de ingewikkeldheid—eufemistisch: ongewoonheid—van (L^A)T_EX weereens probeerde te plaatsen. Dit laatste kwam bij mij op bij het bestuderen van de TUG stijlen. Knap, erg knap, en gezien het doel te knap, naar mijn smaak. Sorry, en met alle respect, dat ik het zeg. Ik heb dat soort zaken al eens eerder meegemaakt in de geschiedenis van de computer. Programmering in de jaren 60 was ongelooflijk knap, maar veelal ook ingewikkeld. Je zou kunnen zeggen dat het een vorm van suboptimalisatie was. Tony Hoare zei het al (geciteerd overigens door Knuth)

‘Premature optimization is the root of all evil.’

Uit die tijd stamt het epitheton ornans monolytische code,³ met als remedie de Reduced Instruction Set Computer, en het modulair, generiek en functioneel programmeren. Later kwam de gebruikersvriendelijkheid ook nog eens om de hoek kijken, en de object-oriented variant van het functioneel programmeren houdt de gemoederen tot op de dag van vandaag bezig. Als je Knuth’s Manmac bestudeert en kijkt wat hij er zoal mee kan—de eenvoud en compactheid van de markup van het T_EXbook en latere publicaties—dan zeg ik laten wij streven naar eenvoud en eerst maar eens proberen de kwaliteit van Knuth te benaderen. Het formatteren via een computer moet het probleem van het publiceren niet vergroten, niet vertroebelen wat je eigenlijk wilde communiceren. Het moet makkelijk en ondersteunend werken. Dit leidt tot mijn boodschap als aftredend voorzitter, aansluitend bij De Vinne, van Krimpen, Schumacher, . . .

laten wij het zo simpel mogelijk houden

en streven naar de eenvoud en kwaliteit van Knuth’s computer typografie.

Een tipje van de sluier wordt opgelicht in mijn BLUE’s serie. Vergelijk eens de eenvoud van BLUE’s Bibliography met BiB_TE_X, mijn tug.ppt met de tugboat stylen, om een praktisch idee te krijgen wat ik bedoel te zeggen. Een ding nog op de valreep. Pas op voor het je laten meesleuren door geïntroduceerde problemen, problemen die er eigenlijk niet zijn, maar opgeroepen door een verkeerde techniek.

Het voorjaar is weer in aantocht, de seizoenen komen en gaan.

²Yannis houdt echter zijn ‘Parametrisation of PostScript fonts through Metafont,’ en vanuit Duitsland is er zonder blikken of blozen ‘Weaving a WEB, the structure and creation of an object network representing an electronic reference work.’

³Ook wel mammoet code genoemd.

Van uw MAPS Editor

Gerard van Nes

Weer een MAPS gevuld!

En met wat voor bijdragen! Ook de redactie kijkt altijd op van de verschillende en veelal zeer actuele onderwerpen die weer geplaatst worden. Natuurlijk het nodige over $\text{\LaTeX}2_{\epsilon}$ en $(\text{\La})\text{\TeX}$ op PC systemen. Verder hypertext zaken, locale $(\text{\La})\text{\TeX}$ gebruikaspecten en ervaringen, anderhalf hoofdstuk uit een onlangs gepubliceerd boek, BLUE's bijdragen uit de bekende hoek, heet-van-de-naald verslagen, etcetc. Het aantal pagina's is wel wat verminderd t.o.v. de vorige twee MAPS uitgaven, doch de inhoud is minstens even interessant! We kregen al wel van de opletende penningmeester het mes al op de keel gezet. Doch na een jubileumjaar mag het best wel eens wat rustiger er aan toe gaan. Al hoewel rustiger? De $4\text{\all}\text{\TeX}$ CD-ROM dient zich al aan als het volgende klapstuk!

Deze MAPS uitgave bevat dus weer voor elk wat wils: zowel de nieuweling, de grote groep van gemiddelde gebruikers, als ook de doorgewinterde documentverwerkende guru komt weer genoeg aan zijn trekken. Deze MAPS is weer goed voor een half jaartje rust aan de kant van de redactie!

Maar wat er voor af ging?

De redactie, bestaande uit Wietse Dol, Jos Winnink, en bovengenoemde heeft deze tweejaarlijkse klus ook nu weer geklaard. Het snuffelen zowel naar mogelijke auteurs als naar algemene en interessante bijdragen (en dat vormt feitelijk al de basis van de MAPS), het vervolgens converteren van de ontvangen artikelen naar de uiteindelijke `maps.sty` layout, het omzetten van de ASCII en *plain* \TeX bijdragen naar standaard \LaTeX , het zo goed als mogelijk verdelen van alle tekst over de bladzijden, de altijd aanwezige (kleine) technische (veelal font) problemen, en een (laatste) 'vluchtige' controle, kost altijd weer de nodige tijd. Maar het zit er weer op.

Overigens zijn we gelukkig enkele jaren geleden niet overgegaan op een `tugboat.sty` of één of andere andere *bestaande* TUG stylefile (zie de bijdrage 'enige suggesties aan de redactie van de MAPS' in MAPS 91.2). De TUGboat redactie heeft helemaal een zware klus. Zouden wij als MAPS redactie niet hebben aangekund.

Een deel van de bijdragen werden in deze MAPS verwerkt via de langzaam opkomende $\text{\LaTeX}2_{\epsilon}$. Jawel aanwezig onder de niet te evenaren 4\TeX ! Overigens zonder nieuwe format binnen 4\TeX gebruiksomgeving was het onmogelijk om de voor u liggende MAPS op tijd klaar te krijgen! De 4\TeX shell heeft de uitgebreide klus weer enigszins kunnen verlichten. Zeker voor (al dan niet complexe) produktie-

omgevingen blijkt 4\TeX de ideale oplossing te zijn.

Tevens is een deel van het editor werk gedaan m.b.v. de $4\text{\all}\text{\TeX}$ CD-ROM. Om de gold-CD te testen heeft Wietse Dol een aantal avonden de CD-ROM gebruikt voor al zijn editors werk. Conclusie na deze zware duur test is dat de CD-ROM al hetgeen wat van hem verwacht wordt *ook daadwerkelijk* kan presteren. De snelheidsverlies vergeleken met een harde schijf is 'slechts' 10-20%

Overigens redactielid Jos Winnink heeft het deze keer wel al te erg te verduren gehad. In de volgende sectie laten we hem even aan het woord.

To convert or not to convert. . .

De redactie van de MAPS staat bij elke uitgave weer voor de vraag: 'Moeten de bijdragen in *plain* \TeX naar \LaTeX worden omgezet worden of niet?'

Op het eerste gezicht ben je geneigd te zeggen laten we de vormgeving vasthouden, die de auteur van de bijdrage heeft bedoeld. En op basis van zijn/haar typografische kennis klaarblijkelijk zinvol vindt. Helaas een blad op een dergelijke manier vullen levert als eindresultaat vermoedelijk op *dat alle artikelen er aardig uitzien, maar dat het geheel een rommeltje is.*

De redactie stelt zich ten doel om leden een *blad* in handen te geven dat bijdragen bevat over een breed terrein $(\text{\La})\text{\TeX}$ beslaand zo mogelijk ingebed in *dezelfde* typografische vormgeving.

Ik hoor u nu al opmerken: *Dus de redactie vindt van zichzelf dat ze meer meer typografische kennis heeft dan de auteurs.*

Het antwoord is eigenlijk ja. En wel omdat het niet zozeer gaat om de vormgeving van de losse bijdrage, maar vooral ook om de vormgeving van het geheel.

Om hier een handvat voor aan de auteurs te geven is de MAPS-stijl ontwikkeld. Weliswaar niet revolutionair, maar toch zodanig dat ze in principe de redactie *die* mogelijkheden geeft om haar werk te doen op een manier, die zij nodig acht.

Voor artikelen aangeboden in \LaTeX is het zeer eenvoudig om deze geschikt te maken (voorzover dat nog nodig is) voor de MAPS-stijl. Artikelen aangeboden in *plain* \TeX vormen echter een geheel ander verhaal! Moet de redactie de auteurs verplichten om hun bijdragen aan te leveren in \LaTeX onder gebruikking van de MAPS-stijl? Of moet de redactie (die ook maar vrijwilligers zijn) de bijdragen zelf maar omzetten, met een poging om de vormgeving te laten aansluiten bij de MAPS-stijl?

Op dit moment kiest de redactie voor de laatste optie. Hetgeen er overigens wel toe heeft geleid dat voor de voor u liggende MAPS één van de redacteuren alleen al in totaal ruim 40 uur gewerkt heeft aan het omzetten van *plain* T_EX bijdragen die uiteindelijk zo'n 50 pagina's omvatten.

Was al dit werk nu nodig? Geen van de bedoelde T_EX artikelen was zodanig dat het niet mogelijk was geweest om ze in L^AT_EX op te zetten. In feite zou het voor de auteurs eenvoudiger zijn geweest om dat wel te doen. Het omzetten bestaat voor een belangrijk deel uit het *verwijderen* van (ongetwijfeld handige en slimme) eigen T_EX constructies van de auteurs. De resulterende artikelen bevatten geen *plain* T_EX instructies meer.

Is het resultaat te zien zult u zich nu afvragen? Ongetwijfeld, maar voor een belangrijk deel komt dat omdat de bijdragen zijn opgenomen in de MAPS en daarom zijn gegoten in een MAPS-stijl-confectiekostuum.

De redactie zou van auteurs dan ook eigenlijk willen vragen om de bijdragen alleen in L^AT_EX te willen aanbieden en *eventueel* gebruik te maken van de MAPS-stijl zodat de werkzaamheden ook voor een redactie bestaande uit vrijwilligers te behappen is.

Dank u!

En de kleine layout aanpassing al gezien?

Jawel, als u goed kijkt zal u een iets bredere bladspiegel zien. Past beter, kan weer meer op een pagina, en geeft soms minder problemen met ondermeer 'verbatim' stukjes. Overigens ook andere (kleine) aanpassingen zijn in de toekomst zeker te verwachten. *Suggesties zijn welkom!*

Trouwens ziet u het resultaat van de nieuwe *NTG PostScript laserprinter*? Begin dit jaar door de NTG aangeschaft en direct al vol op in gebruik bij redactie én NTG secretariaat. Werkt een heel stuk gemakkelijker. Kunnen we eens wat meer doen, zeker ook op het gebied van plaatjes etc.

Thanks to Sebastian Ratz!

De samenwerking tussen editors van gelijksoortige tijdschriften kan er voor zorgdragen dat interessante T_EX/L^AT_EX/etc bijdragen een grotere verspreiding krijgen dat aanvankelijk de bedoeling was.

Sinds de vorige MAPS zijn er ook uitstekende contacten met de editor van het Engelse UK-T_EX tijdschrift **Baskerville**: *Sebastian Ratz*. Uitwisseling vindt nu plaats van bijdragen tussen Baskerville en MAPS!

Ook in de voor u liggende MAPS vindt u, dank zij Sebastian (en de betreffende auteurs die daarvoor hun toestemming gaven!), een aantal prachtige bijdragen, t.w.:

- A brief guide to T_EX assistants,

- Building virtual fonts with 'fontinst',
- Colour slides with L^AT_EX and 'seminar',
- Text to Hypertext Conversion with L^AT_EX2HTML,
- Obtaining T_EX.

Hopelijk heeft deze UK-NL samenwerking een lang leven!

Alle MAPS uitgaven op de 4allT_EX CD-ROM!

De MAPS heeft de CD-ROM gehaald! Aanvankelijk was het de bedoeling alleen de laatste 2 uitgaven van de MAPS (93.1 en 93.2) op de nu reeds veel besproken 4allT_EX CD-ROM te plaatsten, doch spoedig bleek dit medium toch de ideale mogelijkheid te zijn om oude, en veelal niet meer te verkrijgen MAPS uitgaven (een enkel oud exemplaar ligt in een archief ergens op zolder weg te kwijnen), toch weer onder de aandacht van geïnteresseerde lezers te brengen. En niet alleen NTG leden! Ook vanuit het buitenland (vooral Oost Europa) is grote belangstelling voor de elektronisch beschikbare informatie: de MAPS bevat nu eenmaal zowel Nederlandstalige als Engelstalige bijdragen.

Alle MAPS uitgaven, t.w. de nummers 1 t/m 12, staan in ongeveer 160 PostScript files op de CD-ROM. Totaal ruim 25 Mbyte. De verwerking gaf hier ook weer de nodige problemen.

Gelukkig waren wel alle sources nog bewaard gebleven! Alle bestanden (en dat waren er heel wat!) werden eerst overgebracht van een SUN systeem waarop de eerste uitgaven waren gemaakt, naar de 486 PC met het 4T_EX systeem. De nodige SUN specifieke zaken werden verwijderd, tot zelfs uit enkele .vf files (waarin niet-unieke filenamen met meer dan 8 karakters waren opgenomen!). Plaatjes die als 'prints' waren aangeleverd moesten worden gescand. Antieke (1990) tugboat.sty files werden gelukkig nog onder een stoffige VAX gevonden (van compatibiliteit gesproken. . .). Doch uiteindelijk konden de ruim 1600 (!) bladzijden gecontroleerd (lees gepreviewd) worden en in PostScript bestanden worden weggeschreven. De snelheid van de redactie DX2 maakt het vele werk nog enigszins dragelijk. Ook de veel gebruikte 'hotline' met 4T_EX guru Wietse Dol zorgde uiteindelijk voor de totale realisering.

En we gaan gewoon weer door. . .

Niet aan denken, doch over een klein half jaartje is het weer zo ver. Denk aan uw bijdragen, suggesties, ideeën! *Zonder deze kan de MAPS niet verder bestaan.* Een halve pagina zou al mooi zijn. Een beetje meer mag ook. Bijdragen mogen in de steigers staan. Als collega (L^A)T_EX-ers er maar wat aan hebben.

Sluitingsdatum MAPS 94.2 is komende 1 oktober. Iets eerder mag natuurlijk ook!

Jaarverslag NTG

jan–dec 1993

Gerard van Nes

1 Algemeen

Het jaar 1993 was voor de NTG het lustrumjaar. De NTG bestond 5 jaar! Twee druk bezette NTG bijeenkomsten vonden plaats met vele goede lezingen van sprekers van zowel binnen als buiten de NTG. Er verschenen een tweetal uitgebreide MAPS uitgaven (Minutes & Appendices). Medewerking werd wederom verleend aan het $\text{\LaTeX}2_{\epsilon}$ project.

En voor de rest is er het diverse door de leden gedaan waaronder de beschikbaarstelling van de 4\allTeX distributieset en de uitgebreide hulpverlening bij vragen op de TEX-NL discussielijst.

2 Het NTG bestuur

Het NTG bestuur bestond uit de volgende personen:

- C.G. van der Laan, voorzitter,
- G.J.H. van Nes, secretaris,
- J.L. Braams, penningmeester,
- J.J. Winnink, bestuurslid,
- T.A. Jurriens, bestuurslid (tot juni 1993),
- E.H.M. Frambach, bestuurslid (vanaf juni 1993).

Op de NTG bijeenkomst in juni vonden wederom bestuursverkiezingen plaats. T.A. Jurriens en J.L. Braams traden af, waarvan de laatste reglementair. Laatstgenoemde stelde zich ook herkiesbaar. De door het bestuur voorgestelde twee kandidaten, E.H.M. Frambach en J.L. Braams, werden bij acclamatie gekozen.

Philippe Vanoverbeke werd benoemd tot België-commissaris, vooral om de contacten met de Vlaamse leden te verstevigen.

3 Het NTG ledenbestand

Het ledenaantal bleef ook weer in 1993 duidelijk stijgen, zelfs nog sterker dan in 1992. Eind 1993 telde de NTG 232 leden waarvan 33 instituutleden. T.o.v. eind 1992 betekent dit een toename met 39 leden (waaronder 3 instituutleden). De contributie bleef in 1993 wederom ongewijzigd.

De duidelijke samenwerking van NTG met TUG, waarbij het lidmaatschap gecombineerd kon worden overgemaakt, bleek te resulteren in 74 NTG/TUG leden eind 1993; t.o.v. eind 1992 een toename van 26(!) leden.

Helaas bleek bij een aantal leden de toezending van zowel TUGboat als 'TeX en TUG news' zeer lang uit te blijven. De oorzaak hiervan bleek gelukking niet bij de NTG te liggen.

De volledige NTG ledenlijst met aanvullende hardware- en software informatie werd wederom gepubliceerd in de beide verschenen MAPS uitgaven.

4 De NTG bijeenkomsten

Er zijn in 1993 ook weer twee NTG bijeenkomsten georganiseerd welke gekenmerkt werden door een levendige discussie en een hoge mate van informatie-uitwisseling. De voorjaarsbijeenkomst was tevens de eerste NTG lustrumbijeenkomst:

1. Op 10 juni 1993 bij het KNMI te de Bilt. Aanwezig waren 55 leden.
2. Op 18 november 1993 bij Océ te Den Bosch. Aanwezig waren 72 leden.

Op deze bijeenkomsten, met als thema respectievelijk: 'Van Font tot Boek' en '(L^A)TeX Gebruikersomgevingen' vonden de volgende lezingen plaats:

- 'NTG's lustrum: terug- en (interactief) vooruitblik', door Kees van der Laan;
- 'De harmonieeler van het schrift', door Frank Blokland;
- 'Ervaringen met het maken van boeken via \LaTeX ', door Rein Smedinga;
- 'Writing reports with more than 100 people', door Johannes Braams;
- 'Het boek is een heel nuttig hulpmiddel', door Huub van Krimpen;
- ' \TeX : a \TeX workbench for MS-DOS PC's', door Wietse Dol en Erik Frambach;
- 'Scientific Working Environment: $\text{A}^{\text{S}}\text{\TeX}$ ', door Michel Lavaud.

Van bovengenoemde twee vergaderingen (en lezingen) verschenen verslagen met bijlagen (zie MAPS 91.2, 93.1 en 93.2).

Een deel van de lezingen werd gepresenteerd door sprekers, die door de NTG speciaal voor deze gelegenheden waren uitgenodigd.

De leestafels met \TeX -achtige boeken, tijdschriften, brochures en andere aan \TeX verwante documenten, trok wederom de nodige belangstelling. Bij de naajaarsbijeenkomst konden de aanwezigen Addison-Wesley (L^A)TeX/PostScript boeken met reductie aanschaffen.

5 De NTG MAPS

Ook in 1993 verschenen er twee 'jubileum' MAPS uitgaven met ongeveer 475 (!!) goed gevulde bladzijden (in 1992: ongeveer 340) met ondermeer als inhoud: verslagen van nationale en internationale bijeenkomsten, book- en software reviews, artikelen m.b.t. de NTG lezingen, veel tutorial- en ander soortig educatiemateriaal, \TeX / \LaTeX &

PostScript bijdragen, font artikelen, \TeX en PC's, \TeX macro's, de laatste informatie over \LaTeX 3, algemene- en technische \TeX en \LaTeX bijdragen.

De MAPS heeft nu een vaste redactie, bestaande uit Wietse Dol, Gerard van Nes en Jos Winnink.

6 De NTG werkgroepen

De functie van de NTG werkgroepen blijkt toch wat op zijn retour te zijn. Bijdragen komen hoofdzakelijk van individuele leden. De veel gebruikte \TeX -NL discussielijst zorgt dikwijls voor een vorm van 'overleg'.

7 De 4all \TeX distributieset

Op de najaarsbijeenkomst werd de 4all \TeX distributieset, bestaande uit 30 DOS diskettes beschikbaar gesteld. Realisatie hiervan vond plaats dank zij Phons Bloemen, Wietse Dol en Erik Frambach.

Als basis de reeds enige jaren in gebruik zijnde $\mathcal{D}\TeX$ versie, omringd met alle benodigde utilities, vond deze implementatie bij de NTG leden gretige aftrek. Naast het distribueren van het pakketje langs een lijst van adressen in Nederland en België, werd ook kopieerservice geleverd t.b.v. een aantal leden. Bulletin Board gebruikers konden al één dag na de NTG najaarsbijeenkomst de software via modemverbinding ophalen. De 4all \TeX software is tevens beschikbaar gesteld via de bekende RUU ftp-server en CTAN.

8 De NTG CD-ROM

Het begon allemaal met een onschuldige suggestie één dag voor de najaarsbijeenkomst op het NTG bestuursnetwerk. Direct gevolgd met een sterke discussie binnen het bestuur en bij de NTG bijeenkomst een zeer positieve belangstelling. Vervolgens een week weer met ontelbare e-mails zowel binnen het bestuur als met TUG en omliggende zusterverenigingen. Dit alles zou het begin worden van een 4all \TeX CD-ROM welke een half jaar later beschikbaar zou moeten zijn. Veel werk werd hiertoe verzet door het koppel Wietse Dol & Erik Frambach.

Eind 1993 was nog niet te verwachten wat voor een groot impact voor de NTG dit gehele gebeuren wel niet allemaal zou hebben. . .

9 Bestuursactiviteiten

- De voorzitter had vanuit zijn functie zitting in de Board of Directors van TUG.
- De NTG wisselde informatie uit met de bekende LUGs: Exemplaren van de MAPS werden verstuurd. Van een aantal LUG's ontvangt de NTG regelmatig hun periodieken (zie ook leestafel NTG bijeenkomsten).
- Namens de NTG werden in 1993 de volgende bijeenkomsten bezocht:
 - TUG'93 te Aston, Birmingham, UK
Van 26 tot 30 juli werd ondermeer door de voorzitter en de penningmeester deelgenomen aan deze jaarlijkse TUG meeting.
 - CyrTUG'93 te Pereslav'-Zaleskiï, Russia
Van 4 tot 7 oktober werd door de voorzitter deelgenomen aan deze Oost-Europese \TeX gebruikersbijeenkomst.

Van beide bezoeken is uitgebreid verslag gedaan in MAPS 93.2.

- Op 4 februari en 28 oktober vonden telefonische NTG bestuursvergaderingen plaats waarin naast de algemene gang van zaken binnen de NTG, de organisatie van de NTG bijeenkomsten, Public Relation activiteiten, en de gebeurtenissen binnen TUG en andere zusterorganisaties ter sprake kwamen. Daarnaast werd aandacht besteed aan ondermeer bestuursverkiezingen en NTG's toekomst.

10 Diversen

- Van zowel de \TeX -NL fileservers als de \TeX -NL listserver werd in 1993 wederom veel gebruik gemaakt. Het aantal abonnees van de listserver steeg tot ruim 180 (eind 1992: 162). Gedurende 1993 werden in totaal 1047 (in 1992: 1097 mails) verstuurd. Daarnaast werd via de NTG diverse bestanden op de \TeX -NL fileserver geplaatst.
- Het NTG-FGBBS Bulletin Board van Frans Goddijn en Henk de Haan werd door zowel NTG-ers als niet NTG-ers dagelijks bezocht. Het FGBBS bevat op dit moment een indrukwekkende hoeveelheid (\LaTeX) \TeX materiaal.
- Vele leden maakten via e-mail dan wel via ftp dankbaar gebruik van de RUU fileserver van Piet van Oostrum voor het verkrijgen van diverse soorten \TeX materiaal. Daarnaast werd ook gebruik gemaakt van de ftp-server van de Piet Tutelaers (TUE).
- Actieve medewerking werd verleend aan het \LaTeX 2 ϵ project door het \LaTeX 3 team lid Johannes Braams.
- Door NTG leden werd een belangrijk aantal artikelen gepubliceerd in TUGboat, het tijdschrift van de internationale \TeX Users Group:
 - Victor Eijkhout (1993): The bag of tricks, TUGboat 14#1, 23–24.
 - Kees van der Laan (1993): FIFO and LIFO sing the BLUES, TUGboat 14#1, 54–60.
 - Johannes Braams (1993): An update on the babel system, TUGboat 14#1, 60–62.
 - Nico Poppelier (1993): The Donald E. Knuth Scholarship: 1993 Scholar and 1994 announcement, TUGboat 14#2, 102–103.
 - Theo Jurriens (1993): From observation to publication, TUGboat 14#2, 123–126.
 - Nico Poppelier (1993): Review of recent \LaTeX books, TUGboat 14#2, 127–128.
 - Kees van der Laan (1993): Syntactic sugar, TUGboat 14#3, 310–318.
 - Kees van der Laan (1993): Sorting within \TeX , TUGboat 14#3, 319–328.
 - Victor Eijkhout (1993): The bag of tricks, TUGboat 14#4, 424–424.
- Victor Eijkhout was editor van de macro column van de TUGboat.
- Nico Poppelier was lid van de TUG Board of Directors en het 'Knuth scholarship committee'.
- Door Kees van der Laan werd op de TUG94 conferentie te Aston twee lezingen gegeven.

Financieel verslag 1993

Nederlandstalige T_EX Gebruikersgroep

Johannes Braams

Mei 1994

1 Inleiding

Voor U ligt het financieel jaarverslag van de vereniging NTG over het jaar 1994. De financiële ontwikkeling van de vereniging wordt aan de hand van de winst en verliesrekening en de balans besproken.

2 De winst en verliesrekening

In tabel 1 is de winst- en verliesrekening over 1993 weergegeven. De diverse posten worden in de volgende paragrafen kort toegelicht.

<i>Bedragen in guldens</i>	Debet	Credit
Contributie		16.762,30
Sponsoring		
Rente		1.785,88
Evenementen	1.593,55	
Administratie	1.161,06	
KvK en notaris	61,00	
Bestuurskosten	334,50	
Computer faciliteiten		
Mededelingen a/d leden	13.823,57	
Reisbijdragen	2.279,75	
Representatie	2.266,21	
Boekverkoop		927,65
Diversen	1.586,28	
Saldo		3.630,19
	23.106,02	23.106,02

Tabel 1: De winst en verliesrekening over 1993

2.1 Inkomsten

• Contributies

De groei van het aantal leden is in 1993 doorgegaan, waardoor de contributie-inkomsten dit jaar ongeveer f3.700,- hoger zijn dan begroot. Extra inkomsten ontstaan doordat voor elk TUG lid dat door de NTG wordt afgehandeld een vergoeding door TUG wordt betaald.

• Rente

De inkomsten uit rente zijn hoger dan begroot. De inschatting van de hoeveelheid rente inkomsten van de Leeuwrekening was te laag.

• Boekverkoop

In 1993 heeft de NTG een vernieuwde versie van het cursusboek van David Salomon uitgegeven. Op de verkoop van de syllabus wordt een kleine winst gemaakt.

2.2 Uitgaven

• Evenementen

In 1993 vierde de NTG haar eerste lustrum. In het kader daarvan zijn sprekers uitgenodigd voor wie een vergoeding moest worden betaald.

• Administratie

Deze post bevat uitgaven voor het voeren van de administratie zoals portokosten, enveloppen, ordners. Ook de kosten van het verzenden van informatiemateriaal zijn in deze post opgenomen. De kosten zijn bijna twee maal zo hoog uitgevallen als bij het opstellen van de begroting werd geschat.

• KvK en notaris

Deze post omvat de kosten van de inschrijving bij de kamer van koophandel in Groningen.

• Bestuurskosten

Deze post bestaat uit de kosten van telefonische bestuursvergaderingen.

• Mededelingen aan de leden

Deze post bevat de uitgaven gedaan voor het maken, reproduceren en verzenden van het 'verslag met bijlagen'. Voor deze post is bijna f6.000,- meer uitgegeven dan was voorzien. Hiervoor zijn twee oorzaken aan te wijzen. Enerzijds was het ledental hoger dan verwacht, anderzijds zijn de MAPS'en in 1993 vanwege het lustrum extra dik geweest.

• Reisbijdragen

In 1993 zijn bijdragen gegeven voor het bijwonen van de conferentie in Aston door Kees van der Laan en het bijwonen van de CyrTUG bijeenkomst in Moskou door Kees van der Laan.

• Representatie

In deze post zijn de kosten van cadeaus, de kosten van de PR-sets (f 1198,86) en de kosten van de als present exemplaar weggegeven syllabi (f 814,80) opgenomen. Doordat meer syllabi zijn weggegeven dan verwacht is deze post f 966,21 hoger dan begroot.

• Diversen

Het bestuur heeft in 1993 besloten om een bedrag (f1530,28) beschikbaar te stellen voor het bursary fund. Dit fonds wordt gebruikt om mensen die het bijwonen van een conferentie niet kunnen betalen financieel te ondersteunen.

• Saldo

Deze post is geen inkomsten, maar het bedrag dat de vereniging in 1993 te kort is gekomen. Dit tekort is ontstaan door de (eenmalig) hogere kosten van de MAPS,

de hoge PR uitgaven en de bijdrage aan het bursary fund die niet was begroot.

3 De balans

In tabel 2 is de balans per 1 januari 1993 weergegeven.

<i>Bedragen in guldens</i>	Aktiva	Passiva
Giro	23.256,16	
Kas		
Contributies	675,20	
Debiteuren	267,00	
Crediteuren		2.103,85
Kapitaal		12.916,36
	15.020,21	15.020,21

Tabel 2: De balans per 1 januari 1993

<i>Bedragen in guldens</i>	Aktiva	Passiva
Giro	14.078,01	
Kas		
Contributies	675,20	
Debiteuren	267,00	
Crediteuren		2.053,85
Kapitaal		12.966,36
	15.020,21	15.020,21

Tabel 3: De balans per 31 december 1993

In tabel 3 is de balans per 31 december 1993 weergegeven.

De post 'Contributies' betreft leden die hun contributie voor 1993 niet of onvolledig betaald hebben. Het onvolledig betalen wordt veroorzaakt door het geen rekening houden met transfer provisie bij een betaling vanuit België.

De post 'Debiteuren' betreft nog niet betaalde, maar wel geleverde syllabi alsmede geld dat van de TUG rekening moet worden overgeboekt.

De post 'Crediteuren' betreft geld dat naar de TUG-rekening moet worden overgeboekt en een paar nog niet betaalde declaraties.

Het kapitaal is in 1993 afgenomen met f3.580,19; dat was niet begroot, bij de bespreking van de verlies en winst rekening is dit al toegelicht.

4 Samenwerking met TUG

Eind 1991 is een overeenkomst met TUG gesloten die NTG leden 10% korting geeft op het lidmaatschap van beide verenigingen. Daar is door ongeveer de helft van de persoonlijke leden van de NTG gebruik van gemaakt. Vanwege het succes van de regeling is deze in 1993 gecontinueerd.

De regeling geldt nu ook voor mensen die die als instituuts-vertegenwoordiger lid zijn van NTG zij ontvangen 10% korting op het TUG lidmaatschap. Daarnaast is het nu zo dat *alle* lidmaatschapsbetalingen van mensen in Nederland nu via NTG lopen. Ook studenten kunnen van deze regeling gebruik maken, zij krijgen echter geen extra korting vanwege het scherpe studententarief.

Is de overeenkomst vanwege de financiële voordelen een succes, het is gebleken dat de verzending van TUGboat en TTN naar mensen die op deze wijze in 1993 lid zijn geworden van TUG niet geheel vlekkeloos is verlopen. Een aantal leden heeft niet alle numers of zelfs geen enkel nummer ontvangen. Dit moet achteraf gecorrigeerd worden hetgeen tot extra werk heeft geleid. Hopelijk verloopt een en ander in 1994 beter.

5 Conclusie

Voor het eerst in haar bestaan heeft de vereniging NTG meer geld uitgegeven dan zij heeft ontvangen in hetzelfde jaar. Een groot deel van het tekort kan worden verklaard uit het feit dat dit een lustrumjaar was, waardoor extra kosten zijn gemaakt. Wel moet erop gelet worden dat in 1994 de kosten beter in de hand worden gehouden. Afhankelijk van de financiële situatie van de NTG in het najaar zal bij de begroting een voorstel worden gedaan het lidmaatschapsgeld per 1 januari 1995 te verhogen met f10,-.

Use of T_EX* and L^AT_EX within the NTG community

Jos Winnink

Abstract

At the 1993 spring meeting of the Dutch T_EX User Group (NTG) the NTG Board was asked to find out in what way T_EX and L^AT_EX is used by its members. Main motivation for this question was to find out how important funding of the L^AT_EX-3 project for the members of NTG is. Therefore it was necessary to have insight in the use of T_EX and related products/macro packages.

It was decided that the board should held a inquiry among the members and if possible present the results at the fall 1993 meeting. This article describes the results of the survey.

Participation

At the moment the inquiry was held, NTG had about 125 individual members and 62 institutional members. A total of 73 inquiry forms was sent back. Of which 43 forms were filled in by individual members and 30 were filled in by institutional members (institutions were asked to fill out 1 form for the whole institute)

Institutes that returned the forms ranged from 2 to 200–300 T_EX/L^AT_EX users. The table below shows the distribution of the institutes with respect to the number of T_EX/L^AT_EX users.

# of T _E X/L ^A T _E X users	# of institutes
≤ 4	3
5 – 9	8
10 – 19	7
20 – 49	8
50 – 99	1
≥ 100	3

Table 1: Number of T_EX/L^AT_EX users per institute

The use of T_EX and L^AT_EX

The members were asked to indicate what percentage of their T_EX/L^AT_EX work was don by using (plain) T_EX and/or what percentage of their work was done by using L^AT_EX. As can be seen from the table members of the NTG prefer to use L^AT_EX much more than they use (plain) T_EX.

Use of T_EX/L^AT_EX in the (near) future

About 50% of NTGs members think that their use of T_EX/L^AT_EX in the future will grow. No one thinks their use of T_EX/L^AT_EX will decline in the (near) future.

percentage L ^A T _E X	individual members	institutes
	%	%
≥ 90 %	56	73
70 – 89 %	16	10
50 – 69 %	2	7
30 – 49 %	-	3
10 – 29 %	7	3
≤ 9 %	14	3
can't indicate	5	-

Table 2: % of work done with L^AT_EX

future use:	individual members	institutes
	%	%
will grow	51	43
will be stable	40	53
will decline	-	-
don't know	9	4

Table3: Future use of T_EX/L^AT_EX

Why/when is plain T_EX used?

44 % of the individuals and 53 % of the institutional members are using *plain-T_EX* to support or enhance macro packages, like L^AT_EX en (L)A_MS-T_EX. *plain-T_EX* is not used at 22% of the institutes and also not used by 15 % of the individual members.

Local macro packages are being developed by 29 % of the individual members by using T_EX, this use is also found at 19% of the institutes.

*In this and other places the term T_EX is used whenever the system described in the THE T_EXBOOK is meant

Macro packages in use

The following list lists in alphabetical order the distinct macro packages that have been named in the survey.

ARABTeX	BABEL
BEZIER	BIBTeX
DOC/DOCSTRIP	EPSF
EDMAC	Eplain
FANCYHEADINGS	GloTeX
Harvard macro's	(L)AMSTeX
MANMAC	MULTICOL
MultiTeX	NFSS2
-styles	Phyzzx
PICTeX	PSFIG
PSTRICKS	SHADOW
TABLE	TABVERB
TeXinfo	TeXsis
TUGBoat styles	own styles/macro
publishing companies (Elevated and Springer etc.)	

Frontends for \TeX/\LaTeX that were mentioned: Scientific Word and NTS.

What is the importance of $\TeX/\LaTeX/\LaTeX/\LaTeX$ - \TeX ?

The fast majority of the people that answered the survey think that \LaTeX is of great importance for them. For \TeX the importance is felt less. \LaTeX - \TeX is felt to be of hardly no importance.

Nevertheless the great number of uncertainties about the \LaTeX -3 project, this project is regarded as important for the NTG members.

	individual members	institutes
	%	%
(totally) unimportant	44	38
(very) important	44	31

Table 4: Importance of \TeX

	individual members	institutes
	%	%
(totally) unimportant	17	10
(very) important	74	83

Table 5: Importance of \LaTeX

	individual members	institutes
	%	%
(totally) unimportant	71	83
(very) important	-	-

Table 6: Importance of $(L)\LaTeX$ - \TeX

	individual members	institutes
	%	%
(totally) unimportant	21	21
(very) important	57	71

Table 7: Importance of \LaTeX -3 project

Should NTG actively support activities to come to successors for \TeX/\LaTeX ?

The fast majority thinks that NTG should actively support the development for the successors of \TeX/\LaTeX .

support actively?	individual members	institutes
	%	%
yes	78	86
yes, but	7	7
don't know	15	7
no, but	-	-
no	-	-

Table 8: Support actively new developments?

Conclusion

Members of NTG mainly using \LaTeX as their primary interface to the \TeX system. The expectation is that the use of \TeX/\LaTeX will be stable with a tendency to grow in the (near) future. This expectation is partially based on the expectation that the work that is undertaken for the \LaTeX -3 project will succeed.

The making of the 4all \TeX CD-ROM

Erik Frambach

e.h.m.frambach@eco.rug.nl

Abstract

In about one half year the world's first *turn-key* \TeX system on CD-ROM was produced. It contains the ms-dos \TeX workbench 4 \TeX version 3.20 in perhaps the largest form ever, plus many \TeX sources that are valuable to any \TeX user. In this article we present a look behind the scenes, so you get some insight in the course of events that mark the production of NTG's first CD-ROM.

Keywords: CD-ROM, 4all \TeX , 4 \TeX , ISO 9660

1 Just an idea. . .

At the NTG meeting in Den Bosch last year the idea came up to produce a CD-ROM with \TeX stuff. The idea was inspired by the Dutch Unix Users Group who presented its own CD-ROM at their conference a few months before.

Although the majority of the audience did not actually have (access to) a CD-ROM player, it seemed certain to many that they would within a year. So it seemed like a good idea to investigate the possibility of producing a CD-ROM.

As CD-ROMs were and still are becoming more popular, we were not the first to think of producing one. At the international level discussions were going on about the production of a CD-ROM that would contain runnable \TeX installations for a number of operating systems.



In spite of the good intentions, the goal of this project turned out to be too ambitious. The discussions about the set up and the contents led to nothing substantial. Too many people were talking about too many details. An agenda was never made, let alone a time table for decisions to be made.

2 Defining our own way

Still we were convinced that it should be possible to produce a CD-ROM that would be valuable to at least the Dutch \TeX community or the MS-DOS community, because all resources for such a disk are readily available — 4all \TeX , 4 \TeX , MAPS, TEX-NL etc. But a CD-ROM can contain much more, so what else is worth while?

We soon realized that this CD-ROM could easily support packages from other \TeX users groups, and so our decision to produce this CD-ROM was announced to many people all over the world. Anyone who thought he or she had a package that should be on the CD-ROM was invited to contact the 'producers', Wietse Dol and me.

Many, many suggestions came in, and most of them were accepted. Especially the contacts with East-European \TeX users groups was very fruitful.

In the mean time we contacted a cd-manufacturer to discuss technical and financial details. By this time it was clear from the presells that we should produce at least 350 copies.

As the word was being spread, The Kinch Computer Company (remember PC- \TeX ?) got interested. They wanted to become the exclusive dealer for the United States. We turned them down because we preferred distribution through local \TeX users groups. Springer Verlag wanted the rights to reproduce this CD-ROM. Naturally, we turned them down, too. Addison-Wesley, who knew about our activities from the start, failed to produce any proposal for co-operation, in spite of our repeated suggestions.

3 Preparing the data

We wanted to store the data on a network drive for two reasons. First, we don't have any large enough disk available, second, a network drive behaves much like a CD-ROM, i.e. it is read-only. That is a useful characteristic for testing purposes.

Our first estimate was that we could fill up some 200 to 300 MB on the CD-ROM (maximum is about 600 MB). It was no problem to convince the system manager of the Faculty of Economics in Groningen to grant us 300 MB on the network to store all data, but soon we passed that limit. We had to move several packages on the network disk elsewhere to make room for more CD-ROM data.

ISO 9660 defines the standard from CD-ROMs that should be readable on any operating system. The '8.3' rule for file

names comes natural to MS-DOS users, but other rules are more difficult to obey:

- only the letters A–Z, the numbers 0–9 and the underscore ‘_’ may be used in file names, nothing else;
- file extensions are like file names, except that numbers are not allowed;
- directory names cannot have extensions;
- hidden files are not allowed.

4DOS, the powerful command.com replacement, proved very helpful here. In stead of writing a program to check for ISO 9660 compliancy, we could find all deviating files and directories by entering a single command:

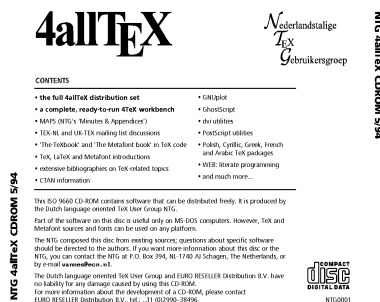
```
DIR * [ !A-Z0-9_ ] * * . * [ !A-Z_ ] * /AD . [ ! ] *
```

The first part lists file names that contain illegal characters; the second lists files with extensions that contain illegal characters; the third lists directories that have extensions. Impressive, isn't it?

We had to rename some files and directories in order to comply to ISO 9660. The spell-checker needed to learn a completely new naming method for dictionaries.

4 Last minute decisions

Should $\text{\LaTeX}2_{\epsilon}$ be on this CD-ROM? No, say the authors, because it is a beta version. Yes, absolutely, say many others. Nobody likes to be a guinea pig, and so few people even attempted to install $\text{\LaTeX}2_{\epsilon}$. If this CD-ROM would contain a complete, installed version they will give it a try. We decided that the user's requests should prevail. On the CD-ROM you will find an installed version, but no installation files. When using \LaTeX , the main menu will show clearly that this is a test version.



The time table we had in mind when we started this project turned out to be a good guess. Of course too many things had to be done on the last days. Michel Lavaud reported that he had forgotten to update some of the files in the French PC-Gutenberg distribution set; Johannes Braams sent us bug fixes to $\text{\LaTeX}2_{\epsilon}$ that forced us to generate new formats; Luigi Bianchi sent updates of his Italian spell-check dictionary; Gerard van Nes sent updates of the MAPS PostScript files, Piet van Oostrum sent the Auc \TeX package, . . . All of this on or just before CD-Day. I guess that is inevitable when you set a deadline.

We deliberately took a month time for the actual production, although theoretically two weeks is enough. After all, it is better to be on the safe side if you are new in this

business. Besides, it gave us some more time to complete the documentation.

5 Transferring the data to cd-rom

We had set a deadline for any contributions to May 2, because May 3 the cd-manufacturer, Euroreseller Distribution BV, would come to collect the data.

The data was transferred to Wim Maasdam's PC through a network card that we installed for the occasion. The transfer took more time than expected because the network performance was rather low that day.

What really puzzled us for a while was that Wim's hard disk showed about 700 files more than the original, although the total number of bytes occupied by all files matched. What is wrong here? Nothing, we deduced. The Novell network doesn't show the '.' and '..' directories. MS-DOS on Wim's computer does show them as directories, but counts them as files! How I love MS-DOS. Unfortunately Wim did not get home in time for the elections that day.

The next day Wim called me to inform me about two files (out of more than 18,000) that did not comply to ISO 9660. That was easily fixed. He would try to make the so-called 'gold-cd' that afternoon. The 'gold-cd' or 'master-cd' is a gold-colored cd from which the normal 'silver-cd's are replicated, after verification of the contents.

There is more. Accidentally we discovered a serious bug in the software that writes the gold-cd. It turned out that the number of files per directory was limited to 650. We had one directory that contains 651 files! Luckily, the software dealer sent an update in just two days.

6 Testing the gold-cd

Immediately after having received the gold-cd, I loaded it into the CD-ROM-player and browsed through the directories. All looked well, and the installation script run just fine. Within 40 seconds I had installed \LaTeX , compiled the sample file, and was viewing the output!

Wietse and I decided we needed two gold-cd's, to be able to test independently. So Euroreseller also sent a copy to Wietse. Unfortunately, they made a terrible mistake. Wietse's CD-ROM could only generate the infamous message 'DRIVE NOT READY. (A)BORT, (R)ETRY, (I)GNORE'. At first Euroreseller wouldn't believe it, but eventually they had to admit that they had sent a completely empty CD-ROM. Deeply embarrassed they personally delivered a new, working gold-cd.

We could find only a few minor glitches, but we considered them important enough to correct in a new gold-cd. Wietse tested this last version and then made the order to produce 600 copies.

7 The final production

Meanwhile Siep Kroonenberg designed the label for the CD-ROM, the back-inlay of the cd-box, and the front page

of the manual. Context BV, who printed the manual, also made a fine offer to produce a full color front-inlay for the cd-box. We gladly accepted, though we would have to insert the inlays by hand. . .

Up until the last day it was unclear what number should be on the CD-ROM. All CD-ROMs must have a unique number, but there is no equivalent to e.g., ISBN for CD-ROMs. We kept it simple — NTG0001 — anticipating more NTG CD-ROM productions.

8 Writing the documentation

Many things had changed since the last edition of the ‘ \TeX & \LaTeX Local Guide’. We rearranged the whole document, revised and updated large parts and added several sections.



We hope the changes and extensions will improve the readability and make the manual even more complete.

Compared to last edition's, the print quality is much higher. The manual is no longer photographically resized, but printed directly on a 600 dpi PostScript printer. In fact, it instantly prints four pages on one A4 page. Then another machine cuts the pages and binds them.

We could print the cover in full color for just the price of the paper as a commercial stunt by Context BV.

9 Looking back

The enthusiastic promotion campaign that the NTG developed almost entirely accounts for the high number of copies we could make. If it weren't for Kees van der Laan and Gerard van Nes, the CD-ROM would never have sold in such quantities. Obviously they believe in the quality of this product, and were able to convince many others, not in the least the Euro \TeX committee that will integrate the CD-ROM in the package that each congress attendant will receive.

Still, we wonder why some users groups showed so little interest. We suspect that political issues are involved here. E.g., the German users group Dante didn't even respond to our first announcement. Although $\text{em}\TeX$ is very popular in Germany, they didn't order any CD-ROMs. However, the German Buchhandlung Hofman ordered 25 copies, and several individuals ordered a copy. Surely more German \TeX users will be interested. Let's wait and see.

One special case is worth discussing in more detail. The \TeX Users Group, TUG for short, at first showed little interest. They were prepared to buy 10 copies; another 40 copies they would keep in store, and possibly return if not sold. Why are they so prudent? Surely they can easily sell at least 100. Remember that Kinch, a commercial company, already requested exclusive selling rights.

Is it money? TUG's financial situation seems weak, though Christina Thiele (the president of TUG) could not give accurate figures. So what is their strategy? Stay put, things will get better? That is silly. They can easily *make* money selling this CD-ROM, can't they see that? It took some serious discussions to convince them, thanks to Michel Goossens (vice-president of TUG) and Sebastian Ratz (UK-TUG). Still, it makes you wonder how TUG is performing at present. . .

10 Looking ahead

The NTG 4all \TeX CD-ROM is the world's first turn-key \TeX system on CD-ROM. Being the first is nice, but naturally some errors and omissions will soon be found.

This CD-ROM represents the current state-of-the-art in \TeX (related) programs for MS-DOS at, say, April 1994. Improved versions of programs will appear and new programs make others obsolete. So what is the expected life time of this CD-ROM?

There are (at least) two possible answers:

1. 25 years (a rough estimate of the life time of CD-ROM technology). \TeX is stable, so who needs updates?
2. 3 months. Soon software updates will appear. Naturally, people want the newest features.

The first answer is tempting, but a bit naive. Nobody can afford to stand still. The second is also naive because new versions are not always better. New bugs and decreased performance (because of 'featuritis') may make you to stick with the good old versions.

We have no plans as yet for producing an update of this CD-ROM, simply because of the time it costs to make one. However, when software updates are released, you can run them in combination with the software on CD-ROM, even within \LaTeX . We will not discuss this in detail here because it is beyond the scope of this article.

We hope that our initiative will encourage others to produce CD-ROMs with \TeX material. I am sure an equivalent CD-ROM for Unix would sell very well. It seems that plans are being made for producing a CTAN CD-ROM. That is good news.

Let me finish here by expressing my gratitude to the NTG that immediately saw the potentials of this project and supported it wholeheartedly. Many, many people contributed to the contents, made valuable suggestions, or promoted the CD-ROM on various occasions. The list of names would become too long, so I will just say thanks to all for your co-operation!

FGBBS: de jonge jarige

Frans Goddijn & Henk de Haan

Postbus 30196

6803 AD Arnhem

BBS tel: 085-217041

(uuencoded files a.u.b. naar FILES@FGBBS.IAF.NL)

goddijn@fgbbs.iaf.nl & haan@fgbbs.iaf.nl

6 april 1994

Abstract

Stonden de eerste zes maanden van het FGBBS in het teken van onafgebroken uitbreiding van file- en berichtengebieden en het telkens vergroten van de beoogde gebruikersvriendelijkheid, het afgelopen halfjaar hebben we met succes gestreefd naar stabilisatie van het systeem, overigens zonder de actualiteit uit het oog te verliezen.

Klaar?

Hoewel een project waarbij een computer wordt ingezet nooit echt af is, begint het er toch naar uit te zien dat het FGBBS is geworden wat het in de ogen van de oprichters moet zijn: een kwalitatief goede en actuele softwarebibliotheek/vraagbaak, voor iedereen toegankelijk, op alle uren van de dag. Aanvankelijk kostte het ons beiden enkele uren per dag om alle verschillende programma's en utilities die samen de FGBBS-dienst uitmaken, in het spoor te houden. Vooral de combinatie van FIDONET- en INTERNET-software, draaiend onder OS/2 besturings-systeem, gaf in het begin veel problemen, die we zelf op moesten lossen aangezien nog bijna niemand anders deze combinatie werkende had. Sinds een paar maanden echter is het systeem verrassend stabiel.

De kwantiteit

De *hoeveelheid* files is ook toegenomen. Een recente lijst met totalen ziet er als volgt uit:

Directory name	Files	Size

Main Directory FGBBS	5	325k
Utilities van algemeen nut	45	6559k
Specifieke (L)A _T E _X utilities	52	7390k
DOC files, L _A T _E X / T _E X manuals & tips	35	2250k
NTG, Nederlandse T _E X Gebruikersgroep	6	36k
TEX-NL, complete collectie berichten	53	2468k
Archief files T _E X Internet-berichten	2	76k
T _E X, L _A T _E X, e _m T _E X standard files	7	2311k
L _A T _E X2 _ε ! The newst and hottest	7	937k
e _m T _E X extensions - uitbreidingen e _m T _E X	9	3114k
Tiny T _E X - piepkleine versie	3	1223k
o _z T _E X, voor Apple Mac	25	8345k
gT _E X, voor 386+ PC's (en DOS box OS/2)	17	7310k
Linux-T _E X Complete versie voor LINUX os	12	5419k
WEB - weef uw eigen T _E X (v. slimme spin)	3	1800k
STYle Files collectie	347	5738k
T _E X shells en editors	43	15174k
Fonts	39	22515k
Font Utilities	17	5846k
DC-Fonts (8 bits extensie op CMR fonts)	6	2258k
BIG T _E X - voor snelle CPU's, veel RAM	4	1020k
DVI utilities, drivers etc.	28	4010k
MetaFont	60	6690k
PostScript	4	424k
Zorba (Grieks)	9	2412k
Other T _E X versions	7	3608k
Bibliography - utils voor bibliografie	8	1111k
Spell Checkers	6	2071k

MusicT _E X - muziekschrift in T _E X	14	2796k
Inzendingen van/voor gebruikers zelf	1	8k
Internet - PC programma's en utilities	7	724k
Offline Reader Software	9	2550k
4allT _E X disk 1	27	1392k
4allT _E X disk 2	13	1414k
4allT _E X disk 3	13	1384k
4allT _E X disk 4	111	1346k
4allT _E X disk 5	18	1361k
4allT _E X disk 6	15	1409k
4allT _E X disk 7	17	1407k
4allT _E X disk 8a	10	1373k
4allT _E X disk 8b	26	1274k
4allT _E X disk 9	15	1403k
4allT _E X disk 10	9	1409k
4allT _E X disk 11	18	1397k
4allT _E X disk 12	20	1408k
4allT _E X disk 13	39	1393k
4allT _E X disk 14	25	1201k
4allT _E X disk 15	11	1399k
4allT _E X disk 16	14	1407k
4allT _E X disk 17	8	1405k
4allT _E X disk 18	8	951k
4allT _E X disk 19	13	1370k
4allT _E X disk 20	15	1404k
4allT _E X disk 21	13	1394k
4allT _E X disk 22	10	1413k
4allT _E X disk 23	12	1382k
4allT _E X disk 24	12	1406k
4allT _E X disk 25	17	1391k
4allT _E X disk 26	29	1244k
4allT _E X disk 27	34	1110k
4allT _E X disk 28	26	1404k
4allT _E X disk 29	16	1357k
4allT _E X disk 30	4	921k

Totals:	1508	170047k

Het files-aanbod van 170 MB betekent een fikse groei die voor een groot deel is te danken aan 4allT_EX. En dan te bedenken dat we een half jaar geleden, toen de honderd megabyte-grens werd overschreden, dachten dat we zo wel ongeveer alles hadden...

Berichten

Wie zijn verhaal over het werken met T_EX kwijt wil, of tips van anderen erover wil lezen, kan terecht in verschillende berichtengebieden. FGBBS.LOKAAL is er voor bellers van FGBBS onderling. Het gebied is voor iedereen toegankelijk, zowel lezend als schrijvend. T_EX.512, verspreid over het FIDONET is ook voor iedereen toegankelijk.

De INTERNET-berichtengebieden zijn voor allen te lezen, maar alleen NTG-leden kunnen (op verzoek) autorisatie krijgen om er zelf ook berichten in te plaatsen. TEX-NL, 4TEX en COMP.TEXT.TEXT behoeven hier geen verdere toelichting. Vooral COMP.TEXT.TEXT is een afdeling waarin vrij grote hoeveelheden post omgaan. Verder zijn er: LATEX-2E, EMTEX en LATEX-L, alsmede voor muzikliefhebbers MUTEK, en voor wie samenvattingen van de COMP.TEXT.TEXT zoekt zijn er TEXHAX en UKTEX. TYPO-L tenslotte is een plek waar bij vlagen schitterende discussies over typografie zijn te vinden, met bijdragen van 'TeX and TUG NEWS'-columnist Peter Flynn.

Hitparade

FGBBS kent twee categorieën gebruikers. Er zijn mensen die interactief, met een communicatieprogramma zoals LittleComm, opbellen en via de menu's van het FGBBS-scherm op zoek gaan naar files en berichten. En er zijn BBS-gebruikers die al dit werk snel en automatisch laten doen door zogeheten 'mailers'. De laatsten komen bijna anoniem binnen, hun software levert een 'verlanglijstje' in bij de mailer van FGBBS en de overdracht van bestanden vindt plaats zonder dat er aan een van beide zijden iemand op hoeft te letten. Vaak zijn deze bellers zelf systeembeheerder van een BBS en geven zij de opgehaalde software weer door aan hun gebruikers.

Hieronder volgt een 'hitparade' van de *interactieve* BBS bellers tussen 15 november 1993 en 1 april 1994. Het sterretje (*) geeft aan dat de bewuste persoon NTG-lid is. In totaal zijn er 165 verschillende mensen langs geweest, waarvan grofweg de helft vaker dan 1 keer. Van deze laatste groep zijn er 17 NTG-lid (uitgaande van de lijst in MAPS 93.2). Het BBS is deze periode 883 keer gebeld, wat neer komt op ruim 6 bellers per dag. De eigen sysops van het FGBBS en 4allTeX medewerker Wietse Dol zijn hier buiten beschouwing gelaten.

Top	Naam	NTG	#
1	John Timmerman	*	169
2	De Coninck Luc	*	143
3	Jan Riesebo		39
4	Gerhard Hoogterp		30
4	Bert Bulder	*	30
6	Jeroen Smulders		28
7	Willem Nijenhuis	*	26
8	Ben Geels	*	18
9	Philippe Vanoverbeke	*	17
9	Joachim Schmitz		17
11	Phons Bloemen	*	15
12	Carel Fenijn		13
12	Berend De Boer		13
14	Jean Prickaerts		10

En dan de hit-parade van bestanden zoals die in de periode van 15 November 1993 tot 1 April 1994 van het FGBBS zijn gesponst. In deze lijst zijn niet alleen de rechtstreekse downloads verwerkt, maar ook de bestanden die via de mailer en via het 4allTeX download-menu zijn overgehaald. De sterretjes geven aan dat het om een bestand

uit de 4allTeX-collectie gaat. In totaal zijn er ruim 3500 bestanden overgehaald, waarvan 60% betrekking heeft op 4allTeX. Er is ongeveer 120 maal een complete disk uit de 4allTeX-collectie gedownload.

File	4	#	File	4	#
FGBBS.ARJ		82	LATEX.ARJ	*	14
ALLINALL.ARJ		24	LAT-A4.ARJ	*	14
NEMO02.ZIP		22	INSTAL01.BAT	*	14
4ALLDOC.ARJ	*	20	F-4ALL1.ARJ	*	14
FGBBS.NEW		19	BABEL-IN.ARJ	*	14
FG-TEX.ARJ		18	ASK.EXE	*	14
WHATS-ON.001	*	18	TXZM220T.ZIP		13
LITTCOM.ARJ		17	UNSETTEX.BAT	*	13
FG-READM.ARJ		17	TFM.ARJ	*	13
LKORT.ARJ	*	17	MAKEINDX.ARJ	*	13
INSTALL.TXT	*	16	LATEXTFM.ARJ	*	13
EMTEXDOC.ARJ	*	16	LATEXNF1.ARJ	*	13
DVISCR.ARJ	*	16	LAT-FMI.ARJ	*	13
DVIHPLJ.ARJ	*	16	FDREAD.EXE	*	13
DVIDOCS.ARJ	*	16	F-4ALL3.ARJ	*	13
4ALLTEX.TXT	*	16	F-4ALL2.ARJ	*	13
JOHN.LOG.DOC		16	DVIDOT.ARJ	*	13
LATEX1.ZIP		15	BIBTXMAN.ARJ	*	13
WHATS-ON.002	*	15	BIBTEX.ARJ	*	13
LKURZ-NL.ARJ	*	15	ARJ.EXE	*	13
DVIDRV.ARJ	*	15	LITTCOM.TXT		12
LATEX2E.ARJ		14	LATEX2.ZIP		12
TEXCHECK.ARJ	*	14	WHATIS.NTG		12
TEX86.ARJ	*	14	PKZ204G.EXE		11
TEX.ARJ	*	14	LATEXDOC.ZIP		11
SET-TEX.BAT	*	14	INSTAL02.BAT	*	11
LATEXSTY.ARJ	*	14	DVIPS.ARJ	*	11

Snelheid

De snelheid van FGBBS is niet alleen op het 80486 DX / 50 mHz moederbord te vinden, maar ook bij cosysop Henk de Haan die niet zelden een nieuwe versie van bijvoorbeeld L^AT_EX₂_ε op het FGBBS plaatste, enige uren nadat deze op het INTERNET was vrijgegeven. Ook heeft Wietse Dol er meermalen, samen met Gerard van Nes, voor gezorgd dat een nieuwe 4allTeXdistributie op zijn minst even snel op het FGBBS was als op de diverse FTP-sites van INTERNET.

Gebruikersmogelijkheden

De beginnende en de incidentele gebruiker van FGBBS zullen voldoende hebben aan de LittleComm software met de bijbehorende utilities zoals John Timmerman die in zijn FGBBS-handleiding heeft genoemd. Wie vaker belt, kan enige leertijd investeren in het ophalen en installeren van de Silver Express software waarmee 'offline' post kan worden gelezen en voorbereid, hetgeen op den duur tijd en communicatiekosten spaart. Bellers die 'stamgasten' worden en die regelmatig een bijdrage leveren aan het FGBBS, in de vorm van hulpvaardige antwoorden op vragen van andere bellers of anderszins, kunnen worden uitgenodigd om een zogeheten 'point' te worden, waarmee opnieuw meer mogelijkheden worden ontsloten.

Hoewel het technisch goed haalbaar is om meer mensen meer faciliteiten te geven zoals een eigen persoonlijk INTERNET-adres en E-mail access via het INTERNET,

houden we dat beperkt. Verdere uitbreiding zou ook hogere kosten met zich meebrengen, en aangezien die kosten tot nog toe zijn gedragen door de beheerders zelf, houden we de koers aan dat alles in principe moet dienen om het \TeX -gehalte van FGBBS te verbeteren. E-mail en andere verdergaande individuele INTERNET-faciliteiten dragen hiertoe niet vanzelf bij. Het ligt eerder voor de hand dat wie de huidige INTERNET-mogelijkheden meer en directer wil exploreren, zelf een aansluiting neemt (bijvoorbeeld bij de Stichting Internet Access).

Floppy-post

Nog steeds is het mogelijk per post files op te halen: stuur hiertoe een aan jezelf geadresseerde en gefrankeerde envelop naar het FGBBS, met daarin voldoende floppies en een *exact verlanglijstje*: welke files uit welke directories moeten erop komen te staan?

Plannen

We zijn vooral van plan het te houden zoals het is: stevig en wendbaar. Binnenkort wordt er een nieuwe, ruimere systeemkast aangeschaft, waarin voor eigen gebruik een CD-ROM-drive wordt bijgeplaatst. We hopen dat de 4all \TeX CD-ROM technische aangrijpingspunten zal bevatten (wat betreft de directory-indeling en files-lijsten) voor koppeling aan het FGBBS. Als dat proefondervindelijk zo blijkt te zijn, dan zou het NTG kunnen besluiten een CD-ROM-drive te bekostigen die 24 uur op alle dagen uitsluitend zal zijn toegewijd aan het via FGBBS beschikbaar stellen van deze NTG-CD-ROM. De hoeveelheid aangeboden files zal daarmee opnieuw behoorlijk toenemen, alles binnen de specialisatie op \TeX -gebied van het FGBBS. De vrijkomende ruimte op de huidige hard disk kan dan deels worden gebruikt voor de actualiteit, zodat daar snel updates kunnen worden geplaatst die nog niet op CD-ROM beschikbaar zijn.

4TEX 3.20 a technical note

Wietse Dol

Landbouw-Economisch Instituut (LEI-DLO)
 P.O.Box 29703
 2502 LS Den Haag
 The Netherlands
 w.dol@lei.agro.nl

Introduction

With the release of the CD-ROM there will also be a new release of the 4TEX workbench (version 3.20). In this note I will try and explain what will be different from the earlier versions (e.g. installed with 4allTEX).

All changes are such that a novice 4TEX user wouldn't see the difference. I.e. the updates are all more or less of a technical nature. This note is therefor for those people who want to know more about the inside of 4TEX and perhaps want to (or need to) change the batchfiles to suit their own preferences. This note will be far from complete but is a first attempt to make some kind of technical reference.

lastdrive=t:

The most important update (we considered this a giant step forward for 4TEX) is that it is now no longer necessary to add the line LASTDRIVE=T to the `config.sys`. This means that 4TEX no longer uses the SUBST or MAP command. This solves many problems for those who want to use 4TEX and also have a network attached to their PC. Instead of using the driveletter T:, we have introduced the environment variable EMTEXDIR. This variable indicates where the root directory of the emTEX system will be (e.g. EMTEXDIR=C:\EMTEX) and is defined in the file `texuser.set`. Because of making 4TEX T: drive independent, the configuration files (`system.set` and `texuser.set`) changed a bit and the Qedit editor macros need to be recompiled. People who install the new 4TEX only need to run a batch file (`inst4tex`) and all the necessary things for a 4TEX installation will be done automatically. So, even the installation becomes simple (on my computer the installation of 4TEX from CD-ROM only takes 45 seconds). People who want to use Graphic Work Shop (GWS) have to install the program and specify the path where to find GWS. This is done by running the installation program `gwsinst.exe`.

emTEX version 11 and DVIxxx version 1.5a

Of course we integrated the new emTEX releases, i.e. beta 11 versions of the TEX compilers and the 1.5a DVI drivers. This means even more printers to select and more advanced features within 4TEX. E.g. the TFM files and TEXINPUT style files can be stored in several subdirector-

ies and with emTEX version 11 it is possible to scan a directory and all its subdirectories simply by adding an '!' to your variable. I.e. `TEXINPUT=C:\EMTEX\TEXINPUT!` will scan `C:\EMTEX\TEXINPUT` and all its subdirectories. Note that this directory expansion is not yet available for the METAFONT programs. With 4TEX we have also made it possible to add directory expansion to several METAFONT variables (e.g. MFINPUT variable), however, this expansion shouldn't be to long. To use this feature you have to set the variable `EXPANDDIR=Y` in the `tex-user.set`. 4TEX will use the batch file `expand.btm` to add all subdirectories of the directories with an '!', and will give a warning when the variable becomes to long.

Utilities

With 4TEX version 3.20 it also becomes possible to add your own utilities to the 'run Utility' option of the main menu. This can be simply done by adding a batch file (.btm) in the directory `\EMTEX\BTM\UTILS`. The first line of the batch file is displayed in the 'utility selection' menu. We advise you to make the first line of your batchfile a comment string (e.g. the first line could look like `: Start my own utility`). After this you can do whatever you like. Do not forget that when you run batchfiles within your batchfile they should be proceeded with the CALL statement. Just take a look at all those utilities already available and making your own utility should be simple. Of course you can add, modify and run user specific utilities by pressing the F5 key anywhere in 4TEX.

Printers and viewers

Adding a new printer or viewer type is really simple. You only have to create a file in the directory `\EMTEX\PRNDEST` specifying the printer and some configuration settings and the printer can be selected from the output menu. Just take a close look at the file `\EMTEX\PRNDEST\EXAMPLE` and you see all the possibilities. After looking at several printer definitions, you can easily make your own. Remember that printer definitions have the file extension .prd. The first line of the printer definition is displayed when you want to select a printer in the output menu. To add a viewer type, just create a file with the extension .vie containing only one line

with the description of the viewer type. This description is displayed in the selection menu and for definitions and variable settings, the printer definition with the same name as the viewer definition is used.

Format files

In the directory `\EMTEX\FORMATS` you will find all kinds of format files that can be generated with `4TEX`. The CD-ROM, for instance, contains 31 format files, ready to use. If you want to add your own format, or change an existing one, here's how to do it. The utility `FORMATS` will look in the `\EMTEX\FORMATS` directory for files with the extension `.for`. These files are used within `4TEX` to generate a format. As an example we will discuss the `lplain.for` file:

```
% lplain           : LaTeX + Babel
% format options   :
lplain
babel.hyp
lfonts.old
\dump
```

The first line contains the name of the format file to be generated (the `%` sign is a comment sign and the whole line after the `%` is ignored by `iniTEX`) and the description

of that format file. In this case `lplain` is the `LaTeX` format and the description will be 'LaTeX + Babel'. The second line contains the `iniTEX` format options necessary to generate the format. This will often be empty but some languages (e.g. Cyrillic or Greek) need specific `iniTEX` settings. After these two lines the rest of the file contains the commands that are given within `iniTEX` to generate the format. In this case the file `lplain.tex` is used for the format definition and after a while this file wants to load the hyphenation file. Because it will not find the specified `hyphen.tex`, `iniTEX` will prompt for another hyphenation file. In this case the file `babel.hyp` will be used to specify the hyphenation patterns to be loaded. After a while `iniTEX` will prompt for the font definition file and `lplain.for` will use `lfonts.old`. After this the command `\dump` will create the format file. After the format file creation, `4TEX` will create a format selection file `\emtex\compiler\lplain.frm` with all the settings necessary for choosing the format. `4TEX` will also store the newly created format file in the appropriate directory (`\EMTEX\TEXFMTS` or `\EMTEX\BTEXFMTS`).

Knowing about the `.for` files also explains how `4TEX` knows which formats are available and how to generate them.

Frequently Asked Questions about \LaTeX

Wietse Dol

Landbouw-Economisch Instituut (LEI-DLO)
 P.O.Box 29703
 2502 LS Den Haag
 The Netherlands
 w.dol@lei.agro.nl

Introduction

Since the presentation of \LaTeX last year, many people who tried to install the system on their PC encountered some problems. All these problems are often solved easily with the help of a few tips (i.e. knowing the correct settings of the variables). We thank all the people who took their time trying to install \LaTeX and reporting their problems. We are not able to test everything and are therefore depending on users to report errors and problems.

To have some kind of quick response on problems we decided to open a discussion list. This discussion list is only for asking \LaTeX , \LaTeX and \LaTeX questions. If you want to subscribe to `4tex@nic.surfnet.nl` send an e-mail to `listserv@nic.surfnet.nl` containing just one line:

```
subscribe 4tex Foo Bar
```

where 'Foo Bar' is your real name. Note that you should *not* send your request for subscription to `4tex@nic.surfnet.nl`, because this is the mailing list itself. In that case your request would be distributed to all subscribers, which will not be appreciated. You are also invited to mail your suggestions to this discussion list. We are proud to say that we already have more than 65 participants from all over the world (yes the only continent we are missing is Africa).

The \LaTeX discussion list is *not* only for specialist. We intend it to be an open list i.e. everyone with whatever question should be able to get a quick and satisfactory answer. So please anyone, ask your questions. You may think they are stupid, but we really like them and also get to know the weak points of our system. This means: points for future updates. . . So keep on asking.

In the sections below we will present some questions and tips from the discussion list, hoping that they are useful to anyone who wants to use \LaTeX .

Questions and tips

1. Out of memory error when previewing a document
2. The `MODE` command in `autoexec.bat`
3. `ansi.com` and the screendisplay of the \LaTeX menu
4. Undocumented TSE feature
5. Double side PostScript Printing
6. `config.sys` and `QEMM`
7. VCPI error messages
8. \TeX capacity exceeded, save set = 1000
9. Some OS/2 remarks

10. Eight magic spaces in `loadpars.btm`
11. Problems with `dvips32`
12. Problems with the T: drive

1 Out of memory error when previewing a document

With \LaTeX there are two programs to view `.DVI` files: `DVISCRC` and `DVISCRS`. `DVISCRC` needs some more memory than `DVISCRS` but is also faster. When you have memory problems you should use `DVISCRS`. With \LaTeX it is possible to select which viewer you would like to use. This is possible by pressing `CTRL-D` in the main menu (i.e. this is a toggle selecting between `DVISCRC` and `DVISCRS`). If you always prefer `DVISCRS` you can add `VIEWSIZE=S` to your `texuser.set` (`VIEWSIZE=` will select `DVISCRC`).

2 The `MODE` command in `autoexec.bat`

I'm using a memory management system resulting in 634 kbytes free RAM. After using \LaTeX I notice a decrease of available free RAM of 27k!! A memory map shows me that after the \LaTeX session, 26k free memory was encapsulated by a resident 4DOS part (0.3k) and a resident `MODE` program (0.5k) resulting in an unavailable RAM part of 27k.

However after including the following line into the `autoexec.bat` file: `MODE LPT1 retry=r` the amount of available free RAM remains now unchanged: 634k *also* after the \LaTeX session.

3 `ANSI.COM` and the screendisplay of the \LaTeX menu

We have had some problems with `ansi.com`. Normally \LaTeX checks if `ansi.sys` is loaded (in the `config.sys`). If not it will load `ansi.com`. However, if you have installed `ansi.com` (e.g. in the `autoexec.bat`) and you have loaded it into high memory (e.g. with the `LOADHI` command) than you are in real trouble. `ansi.com` doesn't recognize itself (when loaded high) and the result of a \LaTeX session will be a screen with a lot of rubbish. So, add `ansi.sys` to your `config.sys` (there are more programs that use `ansi`) or leave it out completely (\LaTeX will load `ansi.com` and unload it when leaving \LaTeX).

4 Undocumented TSE feature

Just for the people who are using the TSE editor: there is an undocumented toggle (CTRL-B in the main menu). This allows you to switch between single error mode and multiple error mode for the compiler. Compiling a document will try to compile the whole document (not stopping for errors) and when errors occur the editor is called and the errors are displayed (one for one).

5 Double side PostScript Printing

Is it possible within \LaTeX to send \TeX PostScript output to a twoside page PostScript printer (not by using the \LaTeX two-pass method). This means starting the PostScript file with:

```

%!PS-Adobe-2.0
statusdict begin true setduplexmode end
%!PS....
%The actual PS file...

```

```

REM twoprtps.btm
@echo off
pushd
cdd %TEXFILES
set OLDPRNDEST=%PRNDEST
iff not exist %FNAME.ps then
  cls whi on bla
  echo The PostScript file %TEXFILES%\FNAME.ps doesn't exist!
  echo We first make the Postscript file and then print it
  echo on a twoside page PostScript printer.
  echo.
  pause
  set PRNDEST=%FNAME.ps
  ctty nul
  %P_PRE
  ctty con
  call %DRIVE\btm\output.btm postscr
  ctty nul
  %P_POST
  ctty con
endiff
iff %@line[%FNAME.ps,1] ne "statusdict begin true setduplexmode end" then
  del _dump_ >& nul
  echo %!PS-Adobe-2.0 > _dump_
  echo statusdict begin true setduplexmode end >> _dump_
  copy _dump_+%FNAME.ps /a _dump_.ps /a
  del _dump_ >& nul
  move _dump_.ps %FNAME.ps >& nul
  set PRNDEST=LPT1
  : change this to specify the correct printer
  call %DRIVE\btm\output.btm %DVICLASS
endiff

:quit
set PRNDEST=%OLDPRNDEST
unset /q OLDPRNDEST
popd

```

6 CONFIG.SYS and QEMM

For QEMM users: take care not to handle `command.com` loading by QEMM: elsewhere all your environment variables and path are lost by leaving \LaTeX and coming back to your starting MS-DOS shell!! This is part of my local `config.sys` file:

```
....
```

Answer: There is always a problem that people want more of a package than the package offers (yes a new update and a lot of money can help but we don't think that is the right answer...) We can not and will not support all possible printer gadgets (It would become a never ending story). However, just to proof how flexible \LaTeX really is I will prompt a solution to your question. Note that this is not \LaTeX and therefore you can use it at your own risk.

To overcome a lot of questions to add utilities we developed a user menu system that can add/remove/run user utilities. This is invoked by pressing the F5 key (did anyone know that? I am interested if anyone uses this utility, I do!) What you want is what I would call a user utility. Therefore press F5 and add the following utility: I would call it `twoprtps.btm` (see below).

I didn't test it (we do not have a postscript printer!) and I suggest you can perfect the batch file yourself. Note that I use (parts) of batch files already used by \LaTeX .

```

; 4TeX does not reset the old DOS
; set environment (incl path)
; using next line:
; SHELL=C:\QEMM\LOADHI.COM /R:2
; C:\DOS\COMMAND.COM C:\DOS\ /e:1536 /p
; SHELL=c:\dos\command.com c:\dos\ /e:1536 /p
....

```

But, using \TeX as your standard shell, you can include the following line into your `config.sys` without any problem: `SHELL=c:\qemm\loadhi.com /r:l /lo c:\emtex\utils\4dos.com`

7 The `TEXINPUT` environment variable

This variable in the \TeX workbench is defined as: `TEXINPUT=t:\texinput!;.c:\texfiles\texinput!` The search sequence is in my opinion different from what a user should expect:

1. the current directory,
2. the user `texinput` directory,
3. the general \TeX `texinput` directory.

Answer:

- `emTeX` always looks in the current directory first. You don't need to include the '.' directory! You even can't get `emTeX` to look in, say, `t:\texinput` first.
- `t:\texinput!` comes first to discourage users to name their style file identical to official ones. We hate to see people use styles like `article` as a \LaTeX style if they have a personal version. Sooner or later it will become incompatible, while other users have no problem running the very same file. Confusion all around!
- It should make no difference that the user input directory is at the end. If your style file has a unique name, `emTeX` will find it anyway. That is the key: unique names prevent confusion. We don't want two or more styles with identical names. However, we can't stop you from using an `article.sty` in the current directory, as explained above – in fact, we use this feature myself for testing purposes.
- Only the TeX compilers (version 3c-beta9 and higher) handle the '!' by themselves – `emTeX` is great! Some other programs use `TEXINPUT` (e.g. `bibtex`, `metafont`), so \TeX expands `TEXINPUT` before running them. For those who like to know how this is done in 4DOS, have a look at `expand.btm`.

8 VCPI error messages

After installing disk 1 through 4 \LaTeX and \TeX were running without problems. After installation of disk 5 (\TeX for 386 and up) I tried to run \LaTeX (or compiling in \TeX) but all I got was the following error message `VIRTUAL MODE NOT SUPPORTED WITHOUT VCPI` What's wrong here?

Answer: The 386 version of the `emTeX` compiler uses the `dos-extender` `EMX`. The documentation of `EMX` explains the above error message as follows:

A VCPI server is required to run `EMX` if the CPU is in virtual mode. You'll get this message if you're using an EMS driver (EMS emulator) which doesn't support VCPI or if you've disabled EMS. Remedy: remove the EMS driver, use an EMS driver which supports VCPI or enable EMS

if it has been disabled (for instance, remove the `NOEMS` keyword from the command line of the EMS driver).

It means that you probably have the following line in your `config.sys`: `device = c:\dos\emm386.exe noems` You will have to remove the 'noems' option, or replace the line by: `device = c:\dos\himem.sys` or, if you have `QEMM`: `device = c:\qemm\qemm386.sys ram`

`QEMM` is a much better memory manager that will give you both XMS and EMS on demand. Highly recommended (though commercial) software.

9 Some OS/2 remarks

\TeX has an excellent feature to create .PK versions of postscript fonts 'on-the-fly' using the `PS2PK` program. This feature can be extremely useful for previewing or printing on a non-PS printer. \TeX however does not contain the standard postscript fonts in a decent format, since these fonts are payware.

People who are using OS/2 have gotten some postscript fonts for free, due to a build in version of the Adobe Type Manager. These fonts however have different names as suggested in `\emtex\ps\fonts\psfonts.inf`, they are located in another directory (`\psfonts`) and do not have associated .AFM files (OS/2 uses a binary file with the extension .OFM for the font metrics).

With some fiddling it is possible to use these fonts in \TeX :

- Change the `PSFONTS` directory in `texuser.set` to the directory where OS/2 stores the postscript fonts (e.g. `d:\psfonts`) and restart \TeX to reload the `texuser.set` file.
- Copy the relevant .AFM files from `\EMTEX\TEXTFONTS\PS`

```
PTMR.AFM  PTMRI.AFM  PTMB.AFM  PTMBI.AFM
PCRR.AFM  PCRR0.AFM  PCRBA.AFM  PCRBO.AFM
PHVR.AFM  PHVRO.AFM  PHVB.AFM  PHVBO.AFM
PSYR.AFM
```

- Put this version of `psfonts.inf` into your Postscript-font directory (you also can modify the existing `\emtex\ps\fonts\psfonts.inf` and copy that version to the Postscript-font directory):

```
* Courier:
rpcrr    -apcrr.afm  cour.pfb
rpcb    -apcrb.afm  courb.pfb
rpcrro   -apcrro.afm  couri.pfb
rpcb0    -apcrbo.afm  courbi.pfb
* Helvetica:
rphvr    -aphvr.afm  helv.pfb
rphvb    -aphvb.afm  helvb.pfb
rphvro   -aphvro.afm  helvi.pfb
rphvbo   -aphvbo.afm  helvbi.pfb
rphvrrn  -aphvr.afm  -E0.82 helv.pfb
rphvbrn  -aphvb.afm  -E0.82 helvb.pfb
rphvron  -aphvro.afm  -E0.82 helvi.pfb
rphvbon  -aphvbo.afm  -E0.82 helvbi.pfb
* Symbol:
rpsyr    -apsyr.afm  symb.pfb
rpsyro   -apsyr.afm  -S0.167 symb.pfb
psyr     -apsyr.afm  symb.pfb
psyro    -apsyr.afm  -S0.167 symb.pfb
* Times:
rptmr    -aptmr.afm  tnr.pfb
rptmb    -aptmb.afm  tnr.pfb
```

```
rptmri -aptmri.afm tnr1.pfb
rptmbi -aptmbi.afm tnrbi.pfb
rptmrrn -aptmr.afm -E0.8 tnr.pfb
rptmrre -aptmr.afm -E1.2 tnr.pfb
rptmro -aptmr.afm -S0.167 tnr.pfb
```

- After these modifications you should be able to use the Courier, Times, Helvetica and Symbol fonts in your documents without using a postscript printer (or using those ugly Ghostscript fonts). However the font metrics will probably be imperfect, since the .AFM and .PFB come from different manufacturers.

10 Eight magic spaces in loadpars.btm

After starting \LaTeX I get the following errors

```
ECHO is OFFx
ECHO is OFFx
```

```
Invalid Parameter "@index[%file,%empty]"
USAGE: IF [not] condition ... es
```

Answer: These errors are caused by an undefined environment variable %EMPTY. Normally, this is set at startup in loadpars.btm and is equal to 8 'spaces', that is, equal to 8 times ALT-255. Some editors (like the Norton Editor) change these hard spaces into normal spaces (i.e. they

change ALT-255 in 32). So, check the variable %EMPTY in the file loadpars.btm and restart \LaTeX .

11 \TeX capacity exceeded, save set = 1000

Sometimes when preparing your document you get (when compiling your code) the 'TEX capacity exceeded' error. This doesn't mean that you made a mistake (it is possible though) but that the default memory settings are not large enough. The default memory settings depend on whether you use BIG or normal \emTeX and the default values can be found in tables (tex.doc and tex386.doc) Just as an example we present the table for normal \emTeX .

Here we see that 'the save size' is by default 600 (and for big=1000). So if this is not enough you can enlarge this (maximum 16000). How?

1. by adding the statement /ms16000 to the variable EM-TEXOPT. So in the \LaTeX main menu give the 'D' for MS-DOS-command and type ESET EMTEXOPT and add the /ms16000
2. in the system.set file there is the variable COMPILEROPT add to this /ms16000, e.g. COMPILEROPT=/MA200 /MS16000

Both should work (do not forget to restart \LaTeX with the /R option).

Option	Removes this TeX error message: TeX capacity exceeded, sorry [...=###]	Range min-max	Default DOS / OS/2
/ma#	parameter stack (macro parameters)	60-1000	60
/mf#	font memory (font metric data)	5000-65500	32766
/mn#	semantic nest size (mode nesting)	20-3000	40 / 100
/mp#	pool size (strings)	20000-65500	50000
/ms#	save size (values saved by grouping)	100-16000	600
/mt#	pattern memory (hyphenation)	5000-65500	10000

Table 1: Memory sizes for \emTeX

12 Problems with DVIPS32

I sometimes have a problem with DVIPS (in about 5% of the cases). Sometimes the task is not finished (all pages are printed to the printer or to file but he gets stuck afterwards) and so now and then he quarrels with QEMM (result: QEMM Exception errors). In both cases the computer has to be rebooted.

Answer: With a 386 machine or higher \LaTeX uses the DVIPS 32 bits program which has his own memory manager. This can cause problems with e.g. QEMM. Solution: I think there will be a new version of dvips32.exe on the CD-ROM. But that is not the straight solution. You can also rename the dvips.exe to dvips32.exe. Now you will use the normal DVIPS instead of the 32 bits version and is therefore a little bit slower (what is slow). Try this and probably no errors will occur. Just remember: \LaTeX is not causing the problem: DVIPS is.

13 Problems with the T: drive

We get a lot of e-mail from people nowadays having problems with the LASTDRIVE=T and especially with the SUBST or MAP commands. The problem arises when someone wants to install \LaTeX on a stand-alone bases with a Novell network attached. Novell claims all the drives from the G: upwards, so it becomes not possible to use the SUBST command. A real problem and not so simple to solve because many of the applications we use need the T: drive.

However, we are always trying to get the most out of the PC and yes I think we are almost ready to have a \LaTeX version that doesn't need a T: drive (and also not a LASTDRIVE=T). We are testing it and looking where problems might occur. It will be on the CD-ROM.

TEX zonder omhaal voor Atari en andere PC's (III)

Robert Best

Abstract

TEXDraw wordt getest. Veel Atari ST's worden gebruikt om elektronisch muziek te maken, op te nemen (MIDI) en af te drukken in muziekschrift. Dat afdrukken kan ook met TEX.

Uitbreidingen van TEX zoals L^ATEX, TEXsis en MusicTEX maken het zetwerk makkelijker en mooier, maar TEX heeft ook meer werkruimte nodig. We maken een BigTEX en een trefwoordenregister voor een lange tekst.

Deze cursus is een herdruk van een serie artikelen uit het blad ST¹, uitgegeven door Stichting ST². Daar zijn de in de tekst genoemde schijffjes te verkrijgen.

Keywords: TEXDraw, MIDITEX, MusicTEX, BigTEX, MakeIndex

1 TEXDraw

TEXDraw is een tekenprogramma, geschreven door Gregor Fritz en Jens Pirnay, dat een tekening omzet in een TEX-file met L^ATEX-tekencommando's. Het is te vergelijken met ZPCAD (zie MAPS 93.1, blz. 66), dat de tekening kan opbergen in een CSG-bestand voor Strunk-TEX. De output van TEXDraw kan met bijna iedere computer verwerkt worden, omdat L^ATEX algemeen verbreid is, terwijl Strunk-TEX beperkt is tot Atari.

TEXDraw staat op PD disk B 113. Het installeren is simpel: `texdraw.prg` en `texdraw.rsc` kopiëren naar een werkschijf. Er zit een duidelijke handleiding bij, `anleitung.dvi`, die zo bekeken en afgedrukt kan worden.

In vergelijking met ZPCAD zijn de tekenmogelijkheden zeer beperkt. Tekenen uit de vrije hand kan niet. Omdat schuine lijnen worden gemaakt met een speciaal font, kunnen ze niet onder elke willekeurige hoek getekend worden. Cirkelstralen kunnen niet elke waarde hebben om dezelfde reden. Ellipsbogen zijn te benaderen met Bézier krommen; L^ATEX heeft dan `bezier.sty` nodig. Voor stippelijntjes is `epic.sty` gewenst dat op schijf B 82 staat.

Het importeren van rastertekeningen is mogelijk, maar nauwelijks bruikbaar. Het PAC-formaat van STAD is om te zetten in een TEX-bestand, al duurt het lang. De IMG → TeX omzetting werkt niet, ondanks de vermelding in de handleiding, maar vrijwel elk raster formaat is met een programma zoals `convert.prg` van Wolfram Heidenreich in een PAC-file te converteren. Erger is dat, als de figuur meer voorstelt dan een paar lijnen, het zo verkregen TEX-bestand erg groot wordt, het inlezen in TEXDraw eindeloos duurt, en L^ATEX het niet meer aankan en 'TeX capacity exceeded' meldt.

Kortom, TEXDraw is alleen bruikbaar voor heel simpele illustraties.

2 MIDITEX

Vroeger tekende een componist noten met stokken en vlaggen, sleutels, kruizen en mollen op notenbalken om de melodie vast te leggen die hij zojuist op de piano gespeeld had. Ook dat handwerk is geautomatiseerd. Met een elektronisch klavier (keyboard, sequencer), aangesloten op de ST, wordt de melodie direct opgenomen als MIDI-bestand, dat vervolgens met een programma (b.v. NOTATOR) automatisch afgedrukt kan worden in muziekschrift. Dat afdrukken kan ook met TEX.

MIDI (Musical Instrument Digital Interface) is voor tonen wat ASCII (American Standard Code for Information Interchange) is voor tekst. Een MIDI-bestand bestaat uit een reeks bytes die o.a. toonhoogte voorstellen, net zoals een ASCII-bestand een reeks bytes bevat die letters en dergelijke weergeven. Er bestaan ook elektronische gitaren, violen, drumstellen enz. om MIDI-files te maken.

Op schijf B 179 staat MIDITEX, geschreven door Hans Kuykens. Dit converteert een MID-file in een TEX-file met MusicTEX-commando's. Op dezelfde schijf staat MusicTEX van Daniel Taupin, waarmee dit TEX-bestand gepreviewd en afgedrukt kan worden.

Copieer `st_m2tex.tos` naar een werkschijf en start deze zelf-uitpakker. Dat levert een folder met o.a. `miditex.ttp` en `example.mid`. Start `miditex` en tik `example -s1 -o1,4,3,2 -b2`

¹Uitgave ST47 en ST 48.

²Stichting ST, Postbus 11129, 2301 EC Leiden, tel. 071-130045. ST is een onafhankelijk tijdschrift van en voor gebruikers van Atari ST computers.

Aria No. 24
(La Création)
Joseph HAYDN
Transcription pour Orgue et Ténor, D. Tausin (1990)

Muziekschrift gezet met MusicTEX

Dat geeft `example.tex`. (Op een Atari TT werkt dit niet, misschien had ik niet de goede versie van `MIDITEX`.) De opties `-s1` enz. worden verklaard in de handleiding van `MIDITEX`; daarover straks meer.

3 MusicTEX

`MusicTEX` is verpakt in `mtex.tos`. Op een harddisk pakt dat prachtig uit door dubbelklikken, als er nog 2,4 Mb ruimte is. Op een schijf van 720 kb werkt dat dus niet. Maar deze TOS-file is eigenlijk een ZIP-file met een aangeplakt uitpakkertje. Met `stzip` van A 545 of B 182 kan je `mtex.tos` als ZIP-file behandelen. Er zit een folder `mtex` in met daarin folders `metafont`, `music`, enz.

De belangrijkste `musicTEX`-bestanden zitten in de folder `music`. Copieer daaruit `musictex`, `musicadd`, `musicnft` en `musictrp` op de TEX-werkschijf, dus bij `tex.ttp`. Copieer de volgende TFM-bestanden uit `mtex\fonts\tfm` naar de TFM-folder op de TEX-werkschijf:

`beamn11`, -13, -16 en -20, `musicn11`, -13, -16 en -20, `slurn16` en -20.

Er moeten nieuwe fonts gemaakt worden, zoals beschreven in MAPS 93.1 blz. 62/63. De fonts in `mtex\fonts\pk` zijn onbruikbaar, omdat ze niet in een folder `fontxxxx` staan. Het zijn geen scherm fonts; wat zijn het wel? Maak op een werkschijf een folder `metafont` en zet daarin de bestanden (`mf.ttp` omgedoopt in) `mf.tos`, `gftopk.ttp` en `plain.bse` van disk B 102. Maak in dezelfde folder een folder `inputs` en copieer daarin `beamn20`, `beamnngen`, `musicn20`, `musicd20` en `musicgen` uit de folder `mtex\metafont`. Start `mf.tos`

en tik de toverspreuk:

```
&plain \mode=atari; mag=1; input musicn20
om musicn20.gf te produceren. Met gftopk.ttp
converteer je dat naar musicn20.pk en dat gaat in de
folder font0096 op de TEX-werkschijf. Maak op de-
zelfde wijze beamn20.pk voor de font0096 folder.
```

Zet ten slotte `example.tex` op de TEX-werkschijf om te previewen. De figuur met de aria van Haydn is gemaakt met `hcreatim.tex` en `creation.tex`.

4 Handleidingen

Met `LATEX` en de zojuist gemaakte fonts is `manual.tex` voor `MIDITEX` af te drukken. Mijn pogingen om de handleiding bij `MusicTEX` af te drukken strandden op de foutmelding 'Font not loaded, not enough room left'. Ik heb de `LATEX`- naar `TEXsis`-commando's omgezet, dat werkt. Deze versie van `musicdoc` staat op de ST disk behorende bij de ST47 uitgave.

-- * --

Het afdrukken van de `LATEX` handleiding bij `MusicTEX` stuit op ruimteproblemen. Met de `TEXsis`-versie op de disk behorende bij de ST47 uitgave gaat het wel goed. Zet `musicdoc.tex` en de index `musicdoc.ind`, het hulpje `musicdoc.aux`, de `MusicTEX` bestanden `musicdex`, `musicnft` en `musicvbm` uit de folder `mtex\music` (zie hierboven), tien TFM-files uit `mtex\fonts` (zie eveneens hierboven) en `texsis.fmt` (zie MAPS 93.2, blz. 114) op de `TEX`-werkschijf. Zo is `musicdoc.dvi` te maken. Maar de index is dan al voorgedrukt.

Op regel 6 van `musicdoc.tex` staat `\def\index#1{}` waarmee de definitie van index leeg gemaakt is. Als je een % voor deze regel zet wordt er wel een index gemaakt. Maar dan krijg je na 16 pagina's één van de betere `TEX`-meldingen:

```
TeX capacity exceeded, sorry ...
If you really absolutely need more
capacity, you can ask a wizard to
enlarge me.
```

Gelukkig staat `BigTEX` op disk B 101, zodat je geen toevenaar in huis hoeft te halen.

5 BigTEX

Waarschuwing: `BigTEX` heeft minimaal 2 Mb RAM nodig. Een harddisk is ook vereist voor wat hier volgt.

I.p.v. `tex.ttp` moeten we `tex.l.ttp` gebruiken, die staat in `bigtex.lzh`. Haal daar ook `initex.l.ttp` uit (zie MAPS 93.1, blz. 60) om een nieuw format te maken. De `l` staat voor 'large'; er zijn ook `h`-versies ('huge', voor 4 Mb), maar dat is niet nodig voor `musicdoc`. Hoe `texsis.fmt` gemaakt wordt staat in MAPS 93.2, blz. 114. Het vergrote format wordt net zo geproduceerd, alleen moet `INITEX` vervangen worden door `INITEX.L`. Gebruik de Engelse `hyphen.tex`. Dat levert `texsis.fml` (let op

de L). Met dit format en nog een hulpje `index.tex` van disk B 182 is `musicdoc.tex` nu echt te verwerken.

Met het originele L^AT_EX-bestand `musicdoc.tex` op B 179 en een vergroot L^AT_EX-format (L) lukt het niet. Misschien gaat het wel met een Huge format, dat heb ik niet geprobeerd.

6 Index

Hoe maak je een index, een lijst van trefwoorden met paginanummers? Daarvoor moet iedere term in de tekst, b.v. `cmr10`, die je in de index wilt opnemen, gemarkeerd worden met `\idx{cmr10}`. Je kunt ook een trefwoord niet in de tekst en wel in de index krijgen met b.v. `\index{cmr font}`. Plain T_EX kent de commando's `idx` en `index` niet; in een plain T_EX-bestand moet je vooraan

```
\input INDEX
```

zetten om `index.tex` in te lezen. T_EX_{sis} zoekt automatisch dit hulpbestand op.

Als je alle trefwoorden gemarkeerd hebt levert `tex.ttp` of `tex.l.ttp` naast het DVI-bestand ook een IDX-file. Met het MakeIndex programma wordt deze file omgezet in een IND-file waarin de trefwoorden alfabetisch gerangschikt zijn. De IND-file kan dan ingelezen worden in het T_EX-bestand met een `input`-commando. In `musicdoc.tex` staat dit commando in het hoofdstuk Index.

7 MakeIndex

`makeindx.ttp`, de Atari versie van het index programma van Pehong Chen, staat op disk B 99. Omdat het voor allerlei tekstformaten bedoeld is, heeft het een 'index style' nodig die het formaat specificeert. Voor plain T_EX en T_EX_{sis} heb ik het volgende bestandje gemaakt en `plain.ist` genoemd:

```
encap 'f'
preamble "{\obeylines "
postamble "\n}"
group_skip ""
item_0 "\n"
item_1 "\n\indent "
```

Start `makeindx.ttp` en tik
`-s plain.ist musicdoc`

om `musicdoc.ind` te maken van `musicdoc.idx`.

De handleiding voor MakeIndex is `ind.tex` dat met een aantal hulpbestanden in `makeindx.lzh` zit op disk B 99. Het is af te drukken met L^AT_EX. Het resultaat is een wetenschappelijk artikel dat het maken van trefwoordenregisters in vele tekstverwerkers behandelt. Hoe je een index style moet maken voor een plain-T_EX bestand vereist wat studie. In tabel 1 en 2 in het artikel staat de syntax, d.w.z. de verklaring van de toverkreten in `plain.ist`.

Een aardig probleem is hoe je iets vet of cursief in de index krijgt. In de IND-file moet b.v. `{\bf Atari}` komen, maar hoe vertel ik MakeIndex dat deze 'entry' niet bij `{\bf}` alfabetisch gesorteerd moet worden maar bij de A? Daarvoor moet in de IDX-file `Atari@{\bf Atari}` komen volgens het artikel (par. 2.3). Wat vóór @ staat wordt gebruikt bij het sorteren en wat er achter staat komt in de IND-file. In de T_EX-file moet je dan `\index{Atari@{\bf Atari}}` tikken. `\idx` werkt hier niet; om 'Atari' in de tekst en in de index te krijgen moet je 'Atari' dus drie maal tikken. Zoiets schreeuwt om een macro:

```
\def\idxbf#1{#1\index{#1@{\bf #1}}}
```

waarmee je alleen `\idxbf{Atari}` hoeft te tikken. Dan vult T_EX overal 'Atari' in waar de parameter #1 staat in

```
#1\index{#1@{\bf #1}}.
```

T_EX zonder omhaal voor Atari en andere PC's (IV)

Mark van Veen

Abstract

Deze keer gaat het over een recent uitgekomen macro-pakket, dat L^AT_EX_{2 ϵ} heet. Dit pakket vormt een krachtige set macro's, waarmee je zonder al te veel van T_EX af te weten goed uitziende teksten kan produceren.

Deze cursus is een herdruk van een serie artikelen uit het blad ST¹, uitgegeven door Stichting ST². Daar zijn de in de tekst genoemde schijffjes te verkrijgen.

Keywords: L^AT_EX_{2 ϵ}

1 Macropakketten

Eén van de sterke kanten van het T_EX-tekstverwerkingssysteem is dat er gemakkelijk krachtige macro's in geschreven kunnen worden. Deze nemen niet alleen veel werk uit handen, maar door anderen geschreven macro's zorgen voor het uitvoeren van ingewikkelde bewerkingen met een simpel commando. Het belang van macro's in T_EX is zelfs zo groot, dat T_EX zonder macro's onbruikbaar zou zijn. Daarom heeft Knuth, de maker van T_EX, een standaard macropakket bij T_EX gemaakt, plain geheten. Iedereen die kale T_EX-commando's gebruikt, zal in veel gevallen commando's uit het plain-pakket gebruiken. Later werden er andere pakketten gemaakt, waaronder de bekendste ongetwijfeld L^AT_EX en A_MS-T_EX zijn, respectievelijk gemaakt door Lamport en Spivak. Maar daarnaast zijn er nog andere pakketten, zoals T_EXsis, Lollipop en T_EXinfo.

In het L^AT_EX-macropakket is geprobeerd het schrijven en het opmaken van de tekst te scheiden. De schrijver kan zich op de structuur van de tekst concentreren, waarbij het vormgeven door L^AT_EX gebeurt. Daarom bevat L^AT_EX veel commando's waarmee de schrijver structurele kenmerken van de tekst aangeeft. Zo zijn daar de part-, chapter-, section-, subsection-, subsection-, paragraph- en subparagraph-commando's die de opdeling van de tekst in hiërarchische eenheden aangeven.

Ook commando's als

```
\footnote{dit is een voetnoot} of
\begin{quote} ... \end{quote}
```

(om een citatie te geven) geven aan wat voor een soort tekst het is, niet hoe het vormgegeven moet worden. Bij de start van een L^AT_EX-tekst wordt aangegeven wat het type tekst is dat geschreven wordt, waarbij eventueel opties meegegeven kunnen worden:

```
\documentstyle[12pt,a4]{article}
```

¹Uitgave ST49.

²Stichting ST, Postbus 11129, 2301 EC Leiden, tel. 071-130045. ST is een onafhankelijk tijdschrift van en voor gebruikers van Atari ST computers.

Ook dit commando heeft invloed op de vormgeving van de tekst. Zo zal met het voorgaande documentstyle-commando een twaalf-punttekst voor a4 papier gemaakt worden, met behulp van de macro's gedefinieerd in het article-pakket.

Het voordeel van deze werkwijze is dat de schrijver zich op de structuur en inhoud van de tekst kan richten en niet afgeleid wordt door de vraag of dit woordje vet of schuin-vet afgedrukt moet worden. Bovendien wordt een consequente vormgeving bereikt, doordat alle kopjes op bijvoorbeeld section-nivo op gelijke wijze worden afgebeeld. Dit voordeel zou in een nadeel omslaan als je als schrijver helemaal geen invloed meer hebt op de vormgeving. Gelukkig is dat niet zo, veel commando's zijn te beïnvloeden met parameters en daarnaast kun je zelf nieuwe commando's maken die uitvoeren wat jij wilt.

2 L^AT_EX_{2 ϵ}

Zeer recent, zo recent dat er momenteel alleen een testversie te verkrijgen is, is het L^AT_EX-pakket geheel vernieuwd en als nieuwe versie uitgebracht: L^AT_EX_{2 ϵ} . Deze versie is gemaakt door een aantal mensen onder wie F. Mittelbach, C. Rowley en R. Schöpf. De eerstgenoemde heeft tesamen met M. Goossens en A. Samarin een boek doen verschijnen over L^AT_EX_{2 ϵ} : *The L^AT_EX Companion*. Later zal ook Lamport een nieuwe editie van zijn boek *A Document Preparation System L^AT_EX* uitbrengen, gebaseerd op L^AT_EX_{2 ϵ} . In de *Companion* staat veel informatie over het veranderen van de standaard-vormgeving van L^AT_EX. Daarnaast bevat het boek veel informatie over allerlei 'style-files', dat zijn kleine pakketjes die een aantal specifieke macro's bevatten. Deze zijn vaak heel handig in combinatie met de standaard L^AT_EX-commando's om allerlei effecten sneller, makkelijker of mooier te bereiken.

Een voorbeeld van een style-file is `multicol.sty`, waarmee ook midden in de tekst meerdere kolommen gezet kunnen worden. Als dat al in een omgeving met kolommen gebeurt, dan worden de kolommen erg smal en levert het wel lelijke regels op.

Het boek van Lamport zal voor degenen die L^AT_EX met de standaard-vormgeving gebruiken handig zijn, daar het niet zo diep op het veranderen van de vormgeving ingaat. Dit boek beschrijft hoe L^AT_EX te gebruiken is.

In L^AT_EX_{2 ϵ} zijn veel verbeteringen en uitbreidingen op het oorspronkelijke L^AT_EX aangebracht. Als eerste valt op dat het startcommando gewijzigd is van

```
\documentstyle
naar
\documentclass.
```

Ontdekt L^AT_EX_{2 ϵ} het `documentstyle`-commando dan neemt het namelijk aan dat het om een oud L^AT_EX-document gaat en schakelt over naar 'compatibility mode'. Het blijft dus mogelijk de oude L^AT_EX-documenten te bewerken. In L^AT_EX_{2 ϵ} worden de style files geladen met het `usepackage`-commando, dat na het startcommando volgt.

De meest in het oog springende verbeteringen zijn het nieuwe font selectie schema (NFSS2) en het inpassen van enkele afzonderlijke macro-pakketten zoals het mathematisch gerichte $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX en het op overheadsheets gerichte SliT_EX. Met behulp van het font selectie schema in L^AT_EX_{2 ϵ} is het veranderen van fonts een stuk makkelijker geworden. Stel dat je een document in een ander font dan het standaard Computer Modern font wilt zetten, bijvoorbeeld in het Pandora-font. In het oude L^AT_EX moest je dan elke fontdeclaratie veranderen. In L^AT_EX_{2 ϵ} is het in feite genoeg om het

```
\DeclareFontSubstitution
commando te gebruiken, waarin panr als nieuw standaardfont voor het document aangegeven wordt. LATEX2 $\epsilon$  zal daarna de benodigde informatie over het font van een speciale file lezen, die in het geval van panr, otlpanr.fd zal heten.
```

Om midden in de tekst van font-familie te wisselen, worden de commando's

```
\renewcommand{\rmdefault}{panr} \rmfamily
gegeven. Het eerste verandert het font dat LATEX2 $\epsilon$  voor de roman tekst neemt en het tweede commando effectueert dat.
```

Daarnaast kunnen met NFSS2 diverse kenmerken van een font apart ingesteld worden. Ben je met een tekst in typewriter bezig en gebruik je een commando om tekst vet te zetten, dan krijg je in L^AT_EX_{2 ϵ} typewriter-vet. In het oude L^AT_EX kreeg je het standaard vette font, Computer Modern Bold.

Het opnemen van vroeger afzonderlijke macropakketten als $\mathcal{A}\mathcal{M}\mathcal{S}$ L^AT_EX en SliT_EX betekent een standaardisatie van L^AT_EX. Nu kun je door de style of document class files voor $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX en Slides te laden de macro's binnenhalen in L^AT_EX_{2 ϵ} . Het voordeel is dat je van de hele machinerie van L^AT_EX_{2 ϵ} gebruik kan blijven maken, terwijl je

toch van de sterke kanten van $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX of Slides gebruik maakt. Bovendien is in een document, door de `usepackage`- of `documentclass`-commando's te zien wat gebruikt wordt. Dat schept duidelijkheid bij anderen die een tekst willen bewerken en printen.

3 Verkrijgen van L^AT_EX_{2 ϵ}

De nieuwste test-versie van L^AT_EX_{2 ϵ} is momenteel in de public domain bibliotheek van de Stichting ST aanwezig. De eerste officiële release wordt dit voorjaar verwacht. Op het moment dat die beschikbaar komt zal de test-versie door de officiële versie vervangen worden. Daarvan zal in ST melding gemaakt worden.

Om L^AT_EX_{2 ϵ} uit te pakken wordt van de zogenaamde `docstrip`-utility gebruik gemaakt. Dat klinkt geheimzinnig, maar het is een instrument om commentaar uit L^AT_EX-files te halen en daarbij de documentatie te genereren voor de uitgepakte file. Allereerst zul je met het decompressieprogramma de file `latex2e.zoo` moeten decompressen in een lege directory op de harde schijf. Dan krijg je een flink aantal files, met een gezamenlijke grootte van ongeveer 2 megabyte. Start nu de T_EX-shell op en ga naar 'environment'. Daar staan de zoekpaden voor T_EX. Voeg dan aan de `inputs`-variabele het pad naar de directory waar `latex2e` staat toe met '`pad`'. Verander de `outputs`- en `format`paden in een pad naar de directory waar `latex2e` staat. Start nu INITEX. In plaats van een INI-file te kiezen ga je naar de directory waar `latex2e` staat en kies je `unpack2e.ins`. Dit verzorgt het uitpakken met de `docstrip`-utility, wat wel 1-2 uur kan duren. Andere INS-files kun je uitpakken door ze, als ware het een tekst, door `tex.prg` te laten bewerken. Zet nu de files met als extensie `cls`, `clo`, `def` en `fd` en de files `slides.ltx`, `docstrip.tex` en `sfontdef.ltx` ergens in de `inputs`-directory weg en zet de files met de extensie `sty` in de `styles` directory. Je moet nu nog met INITEX het L^AT_EX_{2 ϵ} -format genereren. Dat doe je door `latex2e.ltx` te kiezen met INITEX. Zet het gegenereerde format in de `formats`-directory en maak de veranderingen in de `environment` ongedaan (door even uit en in de T_EX-shell te gaan). Alle files die nu nog in de `latex2e` directory staan kun je nu in principe wissen. Je bent dan klaar om L^AT_EX_{2 ϵ} te gebruiken.

Referenties

Goossens M., Mittelbach F. en Samarin A. (1994) *The L^AT_EX Companion*, Addison-Wesley, ISBN: 0-201-54199-8.

De oude versie van Lamport:
Lamport L. (1986) *L^AT_EX: A Document Preparation System*, Addison-Wesley.

L^AT_EX_{2 ϵ} , an overview

Michel Goossens

CN Division, CERN
CH1211 Genève 23
Switzerland
goossens@cern.ch

Abstract

This article gives an overview of the new or extended user commands available with L^AT_EX_{2 ϵ} , the new L^AT_EX release, compared to the previous version L^AT_EX 2.09. After introducing the new preamble commands, the extensions for defining new commands and environments, and handling length and boxes are discussed. The new font selection commands are explained, both for text and math, and it is shown how to easily use different font families. A list of supported class and package files is given and new possibilities for controlling page contents and floats are discussed. Most of this material is described in much greater detail in ‘The L^AT_EX Companion’ [1] and in the second edition of the L^AT_EX Reference Manual [2].

1 Why L^AT_EX_{2 ϵ} ?

Since L^AT_EX became generally available in 1986, its popularity has increased ever since, and many extensions have been developed. Unluckily, these extensions were included in incompatible formats, e.g., ‘standard’ L^AT_EX with and without NFSS, S_LT_EX, $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX, and so on. From the L^AT_EX source alone, it was difficult to determine for which of these (or other) formats a document was composed and, because different sites could have different configuration, document portability was a problem.

Already in 1989 at the Stanford TUG Conference Frank Mittelbach and Rainer Schöpf got together with Leslie Lamport to discuss these (and other) topics and they published their ideas about possible ways to evolve L^AT_EX in TUGBoat [3, 4]. This led a few years later to the start of the long-term L^AT_EX₃ project [5–8].

However, to help end the confusion for the present L^AT_EX users, after a meeting in Spring 1993 between Leslie Lamport and Frank Mittelbach in Mainz, it was decided to release an upgraded version of L^AT_EX, called L^AT_EX_{2 ϵ} , which was officially announced in August 1993 at the TUG Conference at Aston.

Its stated aims are:

- bring all extensions back under a single format;
- prevent proliferation of mutually incompatible dialects of L^AT_EX 2.09;
- NFSS becomes the ‘standard’ font selection scheme;
- make style files like `amstex` (formerly $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX format) or `slides` (formerly S_LT_EX format) into extension packages, all using the same base format;
- add a small number of often-requested features;
- retain the ‘touch and feel’, or the ‘flavour’ of L^AT_EX 2.09.

The first beta version of L^AT_EX_{2 ϵ} was released at the end of 1993, while the first production release is foreseen for ‘Spring 1994’. After that, twice a year (in

‘Spring’ and ‘Autumn’) consolidation releases are planned, in order to keep all versions of the files in synchronization. Bug reports are handled centrally by inviting the users to fill out an electronic form, distributed with the L^AT_EX_{2 ϵ} distribution, and sending it via electronic mail to `latex-bugs@rus.uni-stuttgart.de`. Note that only bug reports referring to the last two releases will be considered. You can also subscribe to the L^AT_EX_{2 ϵ} discussion list on `LATEX-2E@DHDURZ1.BITNET` and post questions (and answers) to that list.

2 Initial and preamble commands

In this section commands which can only be used before or in the preamble will be discussed. The first two below can only appear *before* the `\documentclass` command.

2.1 Initial commands

```
\NeedsTeXFormat{format-name}[release-date]
```

This command, which is normally present in package and class files, can also be useful in user documents to make sure that the file is run with L^AT_EX_{2 ϵ} . Users who try and run it with L^AT_EX 2.09 or plain T_EX will get a reasonably clear error message. An example is

```
\NeedsTeXFormat{LaTeX2e}[1994/02/01]
```

If you want to make sure that your document can be processed at another site, it could make sense that you include all packages and files that your document needs together with the main file. L^AT_EX_{2 ϵ} provides the following syntax to facilitate this

```
\begin{filecontents}{file-name}
  {file-contents}
\end{filecontents}
```

When your document file is run through L^AT_EX_{2 ϵ} the body of each `filecontents` environment is written verbatim

to a file whose name is given as the argument *file-name*. If a file with such a name already exists in any of the directories ‘visible’ to T_EX only an informative message is given, the body of the environment is by-passed, and the file is not replaced.

2.2 Preamble commands

The next commands in the preamble are specifically designed to differentiate L^AT_EX 2_ε documents from those needing L^AT_EX 2.09.

```
\documentclass [option-list] {class-name} [release-date]
```

This command or ‘declaration’ replaces the L^AT_EX 2.09 command `\documentstyle`.

There must be exactly one `\documentclass` declaration in a document, and it must come first (except for the ‘initial’ commands described above).

option-list: list of options that each can modify the formatting of document elements defined in the *class-name* file or in packages loaded with `\usepackage` declarations, as described below.

class-name: name of the class file (file extension `.cls`).

release-date (optional) specifies release date of the class file, using the format `YYYY/MM/DD`. If a version of the class older than this date is found, a warning is issued.

```
\documentstyle [option-list] {style-name} [release-date]
```

This command, which is supported for compatibility reasons, is similar to `\documentclass`, but it loads a ‘compatibility mode’ which redefines certain commands to act as they did in L^AT_EX 2.09 and thus allows you to run your old files unchanged through L^AT_EX 2_ε. Note, however, that in this mode, you *cannot* use any of the L^AT_EX 2_ε extensions described in this article.

You can define new or change existing document elements by loading package files with `\usepackage`, whose syntax is:

```
\usepackage [option-list] {package-name} [release-date]
```

package-name: name of the package (file extension `.sty`); a package can

- define new document elements;
- modify elements defined in the class file;
- extend the range of documents that can be processed.

option-list: list of options, each of which can modify the formatting of elements defined in the package.

release-date: (optional) earliest desired release date of package file (see `\documentclass` command above).

Any number of `\usepackage` are allowed, but L^AT_EX 2_ε makes sure that each package is only loaded once. On

top of processing the list of options specified in the argument *option-list* on its `\usepackage` command, each package also processes the option *option-list* on the `\documentclass` command.

```
\listfiles
```

To help you get an overview of the files read in by your document during processing, you can place a `\listfiles` command in the preamble of your document. This will display the list of all files used at the end of the run.

2.3 Example of document preambles

The following preamble references the `article` class with the (global) options `twocolumn` and `a4paper`, and loads the `multicol` and `babel` packages, the latter with the `german` and `french` options. Other document parameters (e.g., the `textheight`) can also be specified.

```
\documentclass[twocolumn,a4paper]{article}
\usepackage{multicol}
\usepackage[german,french]{babel}
\addtolength{\textheight}{2cm}
\begin{document}
...
\end{document}
```

The following shows three equivalent ways of specifying the loading of packages.

```
\documentclass[german]{book}
\usepackage[german]{babel}
\usepackage[german]{varioref}
\usepackage{multicol}
\usepackage{epic}
```

Somewhat less verbose is:

```
\documentclass[german]{book}
\usepackage[german]{babel,varioref}
\usepackage{multicol,epic}
```

With `german` as global option you can write:

```
\documentclass[a4paper,german]{book}
\usepackage{babel,varioref,multicol,epic}
```

A complex document might look something like the following:

```
\NeedsTeXFormat{LaTeX2e}[1994/05/01]
\begin{filecontents}{varioref.sty}
... % Code for varioref package
\end{filecontents}
\listfiles % print list of files referenced
\documentclass[a4paper,german]{book} % book class
\usepackage{varioref}
\begin{document}
%----- front matter of document
\maketitle
\section*{...} % e.g. section named "Preface"
\tableofcontents % chapter with table of contents
\listoffigures % chapter with list of figures
\listoftables % chapter with list of tables
%----- body of the document
```



```

\part{...}
\chapter{...}
  \section{...}
\chapter{...}
\part{...}
%----- back matter of document
\appendix
\chapter{...} % chapters labelled appendix
\chapter{...}
\begin{thebibliography}
  ... % bibliography entries
\end{thebibliography}
\begin{theindex}
  ... % index entries
\end{theindex}
\end{document}

```

Note that, to ensure that the recipient of the document can process the file correctly, the code of the `varioref` package is shipped with the file inside a `filecontents` environment.

3 Option processing

Options that are specified in the *option-list* argument of the `\documentclass` or `\usepackage` commands are handled as follows:

1. They are first divided into two types, *local* and *global*:
 - for a *class*, the options from its `\documentclass` command are local and there are no global options;
 - for a *package*, the options from its `\usepackage` command are local but the options from the `\documentclass` command are global.
2. The local and global options that have been declared within the current class or package are processed first, normally in their order of declarations, thus their order in *option-list* is irrelevant.
3. Any local options not declared in the current class or package are then processed. For document classes, this usually means that they are ignored, except for this fact being recorded by adding the option to a list of ‘unused options’; they may, of course, be used later since they become global options for every package subsequently loaded. For packages, usually an error message is produced, giving the choice of retyping the option name in case it is incorrect.

Finally, when the `\begin{document}` command is reached L^AT_EX_{2 ϵ} will produce a list of all global options not used by the class or any package file, and issue a warning message for each.

4 Defining new commands and environments

This section and the following describe commands and environments that are used inside the document body (i.e.,

after the `\begin{document}`) command. Let us first look at what is available for defining new commands and environments.

4.1 Defining commands

Commands are defined or redefined in L^AT_EX with:

<pre> \newcommand{\mycom}[narg][default]% {command definition} \renewcommand{\mycom}[narg][default]% {command definition} \providecommand{\mycom}[narg][default]% {command definition} </pre>

The first and second commands show enhancements with respect to L^AT_EX 2.09 by providing the possibility to have an *optional* argument when (re)defining a user command. The presence of such an optional parameter is flagged by the presence of the second optional argument *default*, which specifies the default value of that argument when it is not specified explicitly when the command is used. The last form is useful for general purpose files that are included in a document and over which the user does not always have control (e.g., BIB_TE_X databases). If `\mycom` is not yet defined, the `\providecommand` will act as `\newcommand` and define it, otherwise the existing definition is kept.

The number of arguments, which *includes* the optional argument, is in the range $0 \leq narg \leq 9$. If the command has no arguments, then the `[0]` can be omitted. Inside the *command definition* part, the arguments are referenced as `#1` to `#narg`, the optional argument, if present, being the first one.

For example, compare the following commands, with no, one mandatory, one optional, and one optional and one mandatory argument, allowing the user more freedom in each case.

```

\newcommand{\seq}{x_{0}, \ldots, x_{n}}
\newcommand{\seqm}[1]{#1_{0}, \ldots, #1_{n}}
\newcommand{\seqo}[1][k]{x_{0}, \ldots, x_{#1}}
\newcommand{\seqom}[2][k]{#2_{0}, \ldots, #2_{#1}}
$$\seq\quad\seqm{z}$$
$$\seqo\quad\seqo[1]$$
$$\seqom{y}\quad\seqom[i]{q}$$

```

This gives:

$$x_0, \dots, x_n \quad z_0, \dots, z_n$$

$$x_0, \dots, x_k \quad x_0, \dots, x_l$$

$$y_0, \dots, y_k \quad q_0, \dots, q_i$$

If a command should work both in math and in text mode, special care should be taken in its definition. In L^AT_EX_{2 ϵ} you have the following command:

```
\ensuremath{math code}
```

As its name implies `\ensuremath` ensures that its argument is always typeset in math mode by surrounding it if necessary with `$` signs. For instance, the above can be rewritten as:

```
\renewcommand{\seq}{\ensuremath{x_{0},%
\ldots,x_{n}}}
\renewcommand{\seqm}[1]{%
\ensuremath{#1_{0},%
\ldots,#1_{n}}}
\seq,\quad\seqm{z} or $\seq,\quad\seqm{z}$
```

$x_0, \dots, x_n, z_0, \dots, z_n$ OR $x_0, \dots, x_n, z_0, \dots, z_n$

4.2 Defining New Environments

In \LaTeX 2.09, environments are defined or redefined with the commands:

```
\newenvironment{name}[narg]{begdef}{enddef}
\renewenvironment{name}[narg]{begdef}{enddef}
```

The number of arguments is in the range $0 \leq narg \leq 9$; and, in the case of no parameters, you can omit `[0]`. Inside the definition part, *begdef*, these parameters are referenced as `#1` to `#narg`. If arguments are present, then they are defined when *entering* the environment by specifying them on the command `\begin{myenv}` as shown below.

```
\begin{myenv}{arg_1}...{arg_k}
```

When *exiting* an environment with the command `\end{myenv}` no parameters can be specified. Moreover, the parameters specified with the `\begin{myenv}` command when entering the environment (see above) are no longer available in the definition part *enddef* where you define the actions which should take place when leaving the *myenv* environment.

As with commands, in \LaTeX 2_ε you can now also define environments with an optional (first) argument.

```
\newenvironment{myenv}[narg][default]%
{begdef}{enddef}
```

The default for the optional argument is given between the second pair of square brackets `[default]`. Inside the *begdef* part, which is executed when the environment *myenv* is entered, the optional argument can be accessed with `#1`, while the mandatory arguments (when present) are addressed as `#2` to `#narg`. When the *myenv* environment is used without an optional parameter, `#1` will contain the string specified as `[default]`.

As an example, a variant, `deflist`, of a description environment will be constructed. The `deflist` environment behaves somewhat like a standard \LaTeX description environment if it is used without an optional argument. If an optional argument is specified, then the width of the description label will be put equal to the width of the argument. Thus, by specifying the widest

entry in the list as an optional argument, you ensure that the description parts of all entries line up nicely.

The result below first shows the (default) behaviour of the `deflist` list and then what it looks like when using the optional argument.

```
\newenvironment{deflist}[1][\quad]%
{\begin{list}}{\end{list}}
\renewcommand{\makelabel}[1]{\textbf{##1}\hfil}%
\settowidth{\labelwidth}{\textbf{#1}}%
\setlength{\leftmargin}{\labelwidth+\labelsep}}
\begin{deflist}
\item[First] This is a short term.
\item[Long term] This is a long term.
\item[Even longer term] A very long term.
\end{deflist}
\begin{deflist}[Even longer term]
.....
\end{deflist}
```

First This is a short term.

Long term This is a long term.

Even longer term A very long term.

First This is a short term.

Long term This is a long term.

Even longer term A very long term.

5 Playing with lengths

Lengths can be defined, set and changed by the following commands.

```
\newlength{cmd} \setlength{cmd}{len}
\addtolength{cmd}{len}
\settowidth{cmd}{text} \width
\settoheight{cmd}{text} \height
\settodepth{cmd}{text} \depth
\totalheight
```

The new `\settoheight` and `\settodepth` commands, in analogy with the `\settowidth` command, already present in \LaTeX 2.09, allow one to ‘measure’ the height and depth of some \TeX material. The lengths `\width`, `\height`, `\depth`, and `\totalheight` are also new in \LaTeX 2_ε, and can be used inside the box commands described in the next section.

For ease of reference an overview of \TeX ’s units of length is given below.

sp scaled point (65536 sp = 1 pt)	\TeX ’s smallest unit.	
pt point = $\frac{1}{72.27}$ in = 0.351 mm		
bp big point (72 bp = 1 in), also PostScript point		
dd Didôt point = $\frac{1}{72}$ of a French inch, = 0.376 mm		
mm millimeter = 2.845 pt		u
pc pica = 12 pt = 4.218 mm		┌
cc cicero = 12 dd = 4.531 mm		┌
cm centimeter = 10 mm = 2.371 pc		└

in inch = 25.4 mm = 72.27 pt = 6.022 pc _____
 ex height of a small 'x' for the current font □
 em width of capital 'M' in current font ◻
 mu math mode unit (18 mu = 1 em) ▮

```
\newcommand{\HS}[1][1.]{\hspace{\stretch{#1}}}  
\begin{center}  
left \hfill right\  
left \HS[.5]\fbox{$\frac{1}{3}$}\hfill right\  
left \HS middle \hfill right\  
left \hrulefill\ middle \hrulefill\ right\  
left \dotfill\ right\  
left \dotfill\ \HS[.5] \dotfill\ right\  
left \dotfill\ \HS \dotfill\ right\  
left \dotfill\ \HS[2.] \dotfill\ right  
\end{center}
```

The following lines show how length commands are created, defined, changed, and used. They work, most of the time, both for rigid and rubber lengths.

```
\newlength{\Mylen} Mylen = \the\Mylen
```

Mylen = 0.0pt

```
\setlength{\Mylen}{10mm} Mylen = \the\Mylen  
\setlength{\Mylen}{5mm plus 1mm minus .5mm}  
\par Mylen = \the\Mylen % Use a rubber length
```

```
left right  
left right  
left middle right  
left middle right  
left right  
left right  
left right  
left right
```

Mylen = 28.45274pt

Mylen = 14.22636pt plus 2.84526pt minus 1.42262pt

```
\setlength{\Mylen}{1em} One em is \the\Mylen;  
\addtolength{\Mylen}{1pc} add one pica \the\Mylen.
```

One em is 10.0pt; add one pica 22.0pt.

```
\settoheight{\Mylen}{May} The height is \the\Mylen  
\settoheight{\Mylen}{\Large May} and now \the\Mylen.  
  
\settodepth{\Mylen}{May} The depth is \the\Mylen  
\settodepth{\Mylen}{\Large May} and now \the\Mylen.
```

```
\documentclass{article}  
\usepackage{times}  
\thispagestyle{empty}  
\newcommand{\HRule}{\rule{\linewidth}{1mm}}  
\setlength{\parindent}{0mm}  
\setlength{\parskip}{0mm}  
\begin{document}  
 \vspace*{\stretch{1}}  
 \HRule  
 \begin{flushright}  
 \Huge Geoffrey Chaucer\|[5mm]  
 The Canterbury Tales  
 \end{flushright}  
 \HRule  
 \vspace*{\stretch{2}}  
 \begin{center}  
 \Large\textsc{London 1400}  
 \end{center}  
\end{document}
```

The width is 18.33pt and now 26.39519pt.

The height is 6.73pt and now 9.6912pt.

The depth is 2.18pt and now 3.13919pt.

'Rubber' (variable) lengths are very useful for placing information on the page.

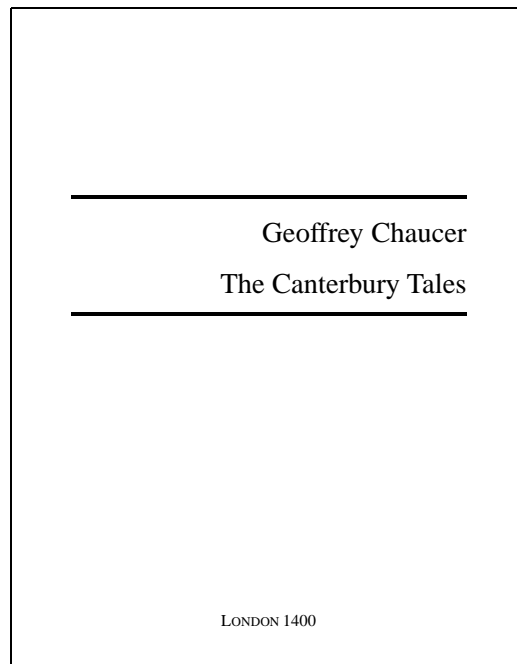
```
\fill
```

This is a rubber length with a natural length of zero. It can stretch to any positive value and its value should not be changed!

```
\stretch{dec_num}
```

This is a more useful rubber length, since `\fill` is equivalent to `\stretch{1}`. More generally, `\stretch{dec_num}` has a stretchability of `dec_num` times `\fill`. It can be used to fine-tune the positioning of text horizontally or vertically.

Examples of the use of these stretchable lengths for controlling the horizontal and vertical page layout are given below.



5.1 Page Markup—Several Kinds of Boxes

Boxes are at the very heart of \TeX 's basic typesetting paradigm, and \LaTeX provides several commands which make it easy to make use of this functionality.

```
\mbox{text}   \makebox[width][pos]{text}
\fbbox{text}  \framebox[width][pos]{text}
```

In addition to centring the text with positional argument [c] (the default), you can position the text flush left ([l]) or flush right ([r]). $\LaTeX_{2\epsilon}$ also offers you an [s] specifier that will stretch your *text* from the left margin to the right margin of the box provided it contains some stretchable space. As already mentioned in the previous section, $\LaTeX_{2\epsilon}$ also allows you to make use of four special length parameters inside the *width* argument of the box commands: `\width`, `\height`, `\depth`, and `\totalheight`. These parameters specify the natural size of the *text*, where `\totalheight` is the sum of `\height` and `\depth`.

The examples below show how these various parameters are used to control the layout in the box. Note that use is also made of the `calc` package, which allows arithmetic operations in the arguments of the commands.

```
\framebox{A few words of advice}           \par
\framebox[\width + 6mm][s]{A few words of advice}
\par \framebox[1.5\width]{A few words of advice}
```

A few words of advice

A few words of advice

A few words of advice

```
\rule[lift]{width}{total height}
```

Rules come in handy for controlling the height of a box. Together with the new \LaTeX commands for measuring the height and depth of your boxes, they allow you to perform micro-typographic adjustments for tuning the visual presentation of your document elements.

```
\newsavebox{\Maybox}\savebox{\Maybox}{\Large May}
\newlength{\Mdp}\settodepth{\Mdp}{%
  \usebox{\Maybox}}
\newlength{\Mht}\settoheight{\Mht}{%
  \usebox{\Maybox}}
\addtolength{\Mht}{\Mdp}

\framebox[1.6\width+1em][s]{\usebox{\Maybox}}
\quad
\framebox[1.6\width+1em][s]{\usebox{\Maybox}%
  \rule[-2\Mdp]{0mm}{2\Mht}}
```

May

May

Zero-width boxes are also useful in other circumstances.

```
\begin{center}
A centred sentence.\makebox[0cm][l]{\textsuperscript{123}}\
Some more text in the middle. \
\makebox[0cm][r]{\textsuperscript{321}}A centred sentence.\
\end{center}
\noindent\makebox[0cm][r]{\(\Leftrightarrow\)}%
As seen in the margin of the current line, boxes
with a vanishing width can stick out in the margin.
```

A centred sentence.¹²³

Some more text in the middle.

³²¹A centred sentence.

⇒ As seen in the margin of the current line, boxes with a vanishing width can stick out in the margin.

5.2 Moving boxes

Boxes can be moved up or down by the command:

```
\raisebox{lift}[depth][height]{contents}
```

The simple example below shows its principle of use.

```
\begin{flushleft}
x11x \raisebox{-1ex}{downward} x222x \
x333x \raisebox{1ex}{upward} x444x \[1em]
x11x \raisebox{-1ex}[0cm][0cm]{downward} x222x\
x333x \raisebox{1ex}[0cm]{upward} x444x
\end{flushleft}
```

x11x downward x222x
x333x upward x444x

x11x downward x222x
x333x upward x444x

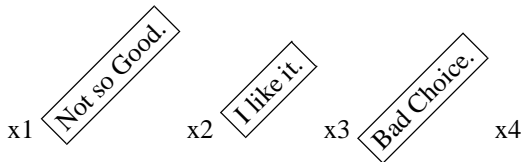
A more useful example is the generation of text ‘between’ two rows in a table (by ‘hiding’ the boxes’ content from \TeX .)

```
\begin{center}
\begin{tabular}{|c|c|c|} \hline
& \multicolumn{2}{c}{title} \\ \hline
\raisebox[1.5ex][0cm][0cm]{100}
& A & B \\ \hline
20000000 & 10 & 10 \\ \hline
\end{tabular}
\end{center}
```

100	title	
	A	B
20000000	10	10

Finally, when your printer driver allows it, you can rotate boxes. In this case the use of the various box dimension parameters becomes apparent.

```
\newcommand{\DoT}[1]{\begin{turn}{45}#1\end{turn}}
x1 \DoT{\fbbox{Not so Good.}} x2
\DoT{\raisebox{\depth}{\fbbox{I like it.}}} x3
\DoT{\raisebox{-\height}{\fbbox{Bad Choice.}}} x4
```



5.3 Placing parboxes and minipages

In LaTeX 2.09, boxes, which can contain more than one paragraph are defined as follows.

```
\parbox[pos]{width}{text}
\begin{minipage}[pos]{width}
text
\end{minipage}
```

A simple example of its use is the following

```
\parbox{.3\linewidth}{This is the
contents of the left-most parbox.}
\hfill Centerline \hfill
\parbox{.3\linewidth}{This is the right-most
parbox. Note that the typeset text looks
sloppy because LaTeX cannot nicely balance
the material in these narrow columns.}
```

This is the contents of the left-most parbox.

Centerline

This is the right-most parbox. Note that the typeset text looks sloppy because LaTeX cannot nicely balance the material in these narrow columns.

5.4 Generalized parboxes and minipages

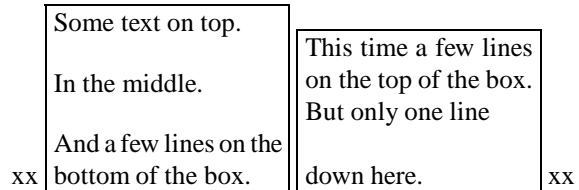
Sometimes it is helpful to predefine the vertical dimension of a paragraph box. For this LaTeX 2 ϵ has additional optional arguments for minipage and parbox.

```
\parbox[pos][height][inner-pos]{width}{text}
\begin{minipage}[pos][height][inner-pos]{width}
text
\end{minipage}
```

The *inner-pos* determines the position of *text* within the box. It can be t, c, b, or s. If not specified, the value of *pos* will be used. You can think of *height* and *inner-pos* as the vertical equivalent of the *width* and *pos* arguments of a `\makebox`. If you use the *s* position the *text* will be vertically stretched to fill the given *height*. Thus, in this case you are responsible for providing vertically stretchable space if necessary using, for example, `\vspace` or `\vfill` commands.

As with the other box commands you can use `\height`, `\totalheight`, and so on to refer to the natural dimensions of the box when specifying the optional argument.

```
xx \fbox{\parbox[b][1.5\height][s]
{30mm}{Some text on top. \par\vfill
In the middle. \par\vfill
And a few lines on the
bottom of the box.}}
\fbox{\parbox[b][\height+\baselineskip][s]
{30mm}{This time a few lines on the
top of the box. But only one
line \par\vfill down here.}} xx
```



5.5 Manipulating Boxed Material

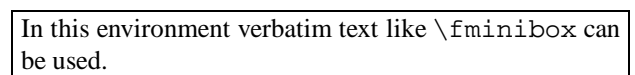
Material can be typeset once and then stored inside a named box, so that its contents can be retrieved later. LaTeX offers the following commands for dealing with this situation.

```
\newsavebox{cmd} declare box
\sbox{cmd}{text} fill box
\savebox{cmd}[width][pos]{text} fill box
\usebox{cmd} use contents
\begin{lrbox}{cmd} fill box
text
\end{lrbox}
```

Note that the environment `lrbox` is an addition in LaTeX 2 ϵ . *cmd* should be a box register previously allocated with `\newsavebox`. The environment `lrbox` will save *text* in this box for later use with `\usebox`. Leading and trailing spaces are ignored. Thus, `lrbox` is basically the environment form of `\sbox`. You can make good use of this environment if you want to save the body of some environment in a box for further processing. For example, the following code defines the environment `fminipage` that works like a `minipage` but surrounds its body with a frame. Note the use of the optional argument for controlling the width of the boxed minipage, and the fact that verbatim material can be used inside. To be able to do the arithmetic operations you will also need to have the `calc` package loaded.

```
\newsavebox{\fminibox}
\newlength{\fminilength}
\newenvironment{fminipage}
[1][\linewidth]% default width is \linewidth
{\setlength{\fminilength}%
{#1-2\fboxsep-2\fboxrule}%
\begin{lrbox}{\fminibox}%
\begin{minipage}{\fminilength}}
{\end{minipage}\end{lrbox}%
\noindent\fbox{\usebox{\fminibox}}}

\begin{fminipage}
In this environment verbatim text like
\verb=\fminibox= can be used.
\end{fminipage}
```



```
\begin{fminipage}[.5\linewidth]
  ....
\end{fminipage}
```

In this environment verbatim text like \fminibox can be used.
--

5.6 For hackers only: a list with two optional parameters

What if we want to define a command or environment with, e.g., *two* optional arguments? Suppose we want a list where we are able to specify not only the width of the label, but also whether the list should be ‘dense’ or not, i.e., we want a syntax like:

```
\begin{Description}[<margin>][<style>]
```

In this case we use a trick (and the packages `calc` and `ifthen`) and introduce a multiple-step definition. The example also shows how one can parameterize the various typographic parameters for the list so that they can be controlled more easily (e.g., the alignment of the label, its font, and the width of the margin).

```
\newcommand{\Descriptionlabel}[1]{%
  \mbox{\Descriptionfont #1}\hfil}

\newcommand\Descriptionfont{\itshape}
\newcommand\Descriptionmargin{}

\newenvironment{Description}[1][\kern\leftmargin]
{\renewcommand\Descriptionmargin{#1}\xdescription}
{\endlist}

\newcommand{\xdescription}[1][normal]{%
  \list{}{\settowidth{\labelwidth}%
    {\mbox{\Descriptionfont\Descriptionmargin}}%
    \setlength{\itemindent}{0pt}%
    \setlength{\leftmargin}{\labelwidth+\labelsep}%
    \let\makelabel\Descriptionlabel
    \ifthenelse{\equal{#1}{compact}}%
    {\setlength{\itemsep}{0pt}%
     \setlength{\topsep}{.5\topsep}}}%
  }}

```

```
Text before text before text before text before
\begin{Description}
  \item[First] This is the first item in the list.
  \item[Veryyy long] This is a veryyy long item.
  \item[] This is an empty item.
\end{Description}
```

```
\begin{Description}[Veryyy long]
  ...
\end{Description}
```

```
\begin{Description}[Veryyy long][compact]
  ...
\end{Description}
```

Text before text before text before text before

First This is the first item in the list.

Veryyy long This is a veryyy long item.

This is an empty item.

First This is the first item in the list.

Veryyy long This is a veryyy long item.

This is an empty item.

First This is the first item in the list.

Veryyy long This is a veryyy long item.

This is an empty item.

6 Font commands—an overview

This section covers the user commands in $\text{\LaTeX}2_{\epsilon}$ for specifying fonts, both in text as in mathematics. We also mention some of the more popular fonts packages and say a few words on compatibility with $\text{\LaTeX} 2.09$.

The first question you can naturally ask yourself is why new font commands were introduced at all. To answer this question let us mention that $\text{\LaTeX} 2.09$ font commands had a few idiosyncrasies:

- their syntax, i.e., `\it foo` rather than `\it{foo}`, which is unlike the syntax of (most) other \LaTeX commands (safe the size-changing series), which are specified with arguments;
- the font commands were not *orthogonal*, e.g., `\bf\sff` produces medium-weight sans, i.e., only the inner font command is honored;
- some fonts substitutions were taking place ‘behind our backs’, e.g., `\tiny\tt` produces tiny roman, since it was assumed that at such a small size the difference is hardly visible, so that one can as well use a font already loaded;
- italic corrections must be introduced by hand, e.g., one has to write `\em my text\`, and even this is not correct in all circumstances.

$\text{\LaTeX}2_{\epsilon}$ addresses these problems by introducing the following new text font commands:

```
\textmd{This is medium text}
\textbf{This is bold text}
\textup{This is upright text}
\textit{This is italic text}
\textsl{This is slanted text}
\textsc{THIS IS SMALL CAPS TEXT}
\textrm{This is roman text}
\textsf{This is sans text}
\texttt{This is typewriter text}
```

Plus `\emph{This is emphasized text}`.

The size changing commands remain unchanged (i.e., `\large`, `\scriptsize`, etc. are still valid).

These commands do not have the problems of the $\text{\LaTeX} 2.09$ commands, because:

- their syntax is the same as for the other \LaTeX commands;
- `\textbf{\textsf{text}}` produces bold sans;

- `\tiny\texttt{text}` produces tiny typewriter;
- `\emph{text}` does *not* need `\/`.

Note that there are still some restrictions, for instance, `\textbf{\texttt{text}}` produces medium typewriter for lack of a bold Computer Modern typewriter font, but at least L^AT_EX_{2 ϵ} warns you about the substitution.

In the area of math fonts, L^AT_EX_{2 ϵ} provides the following new commands:

```
\mathnormal{This is normal math italic}
\mathcal{MATH CALLIGRAPHIC}
\mathrm{This is roman in math}
\mathbf{This is bold in math}
\mathsf{This is sans in math}
\mathit{This is text italic in math}
\mathtt{This is typewriter in math}
```

Note that these commands do not work outside mathematics.

It is now relatively easy (if you have the fonts) to replace Computer Modern with other font families. Various packages for popular fonts are already available, for example:

- In the area of PostScript fonts `\usepackage{times}` provides Adobe Times, `\usepackage{palatino}` Adobe Palatino, `\usepackage{lucidbr}` Y&Y's LucidBright and LucidaNewMath, etc.;
- `\usepackage{amssymb}` provides the AMS fonts;
- `\usepackage{pandora}` allows you to use the Pandora fonts;
- `\usepackage{euler}` lets you experiment with Hermann Zapf's Euler font family.

L^AT_EX 2.09's old font commands (`\rm`, `\bf`, etc.) are still available in L^AT_EX_{2 ϵ} , but they are *not* part of the 'kernel'. They are now defined in the document class files, where the definitions of the size changing commands, like `\huge`, `\tiny`, have always resided. It is thus up to the document designer to define how the old font commands behave. Note, however, that for the 'standard classes' (`article`, `book`, etc.) the old font commands behave as they did in L^AT_EX 2.09.

One more word about about L^AT_EX 2.09 compatibility. A document beginning with `\documentstyle` is run in *compatibility mode*, which emulates L^AT_EX without NFSS. If you want to emulate L^AT_EX with NFSS you should say:

```
\documentstyle[newfont]{...}
```

7 Standard Classes in L^AT_EX_{2 ϵ}

This section discusses the files that come with the L^AT_EX_{2 ϵ} distribution and lists some of the packages which are already adapted to L^AT_EX_{2 ϵ} .

Files associated to L^AT_EX_{2 ϵ} are characterized by the extensions:

`name.cls` for class files;
`name.clo` for external option files;

`name.sty` for package files
`name.cfg` for runtime configuration files

The 'standard' document classes distributed with L^AT_EX_{2 ϵ} are `article`, `report`, `book`, `letter`, `slide`, `proc`, and `ltxdoc`. Below, we say a few words about each one of them.

article, report, book

- they behave like the old L^AT_EX 2.09 styles;
- `twocolumn` and `openbib` are now internal options;
- a set of new internal options was added: `a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`, `landscape`.

letter

- it behaves like the old style;
- it has a set of new internal options: `a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`.

slide

- it behaves like the old style, but used with L^AT_EX_{2 ϵ} ;
- it supports local font configuration by looking for `sfonts.cfg`;
- it has a set of new internal options: `a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`, `landscape`;
- the option `twocolumn` is not supported.

proc

- it is no longer an option but a *document class*;
- it is built on the `article` class;
- it disallows options `a5paper`, `b5paper`, `onecolumn`, `titlepage`.

ltxdoc

- it is used to format the L^AT_EX_{2 ϵ} source code;
- it is built on `article` and requires the `doc` package;
- it looks for the configuration file `ltxdoc.cfg`;
- it defines the commands `\DocInclude` and `\GetFileInfo`;
- it disallows the option `a5paper`.

Presently the following packages are available:

- `ifthen`, for building control structures. It provides on top of `\ifthenelse` and `\whiledo` available previously with L^AT_EX 2.09, the new commands `\newboolean`, `\setboolean`, and `\boolean`.
- `makeidx`, `showidx`, to help you make indexes.
- `doc`, `shortvrb`, for generating class and package file documentation.
- `oldfont`, `newfont`, for compatibility with L^AT_EX 2.09 and version 1 of the NFSS.
- `latexsym` L^AT_EX_{2 ϵ} no longer loads the `lasy` fonts by default; if needed they become available by loading this package. Note that they are not necessary when either `amsfonts` or `amssymb` is used.
- `exscale` allows for different math extension fonts.
- `eufrak` and `euscript` give access to the Euler fraktur and script alphabets, `oldgerm` to Haralambous' beautiful old German fonts, while `pandora` allows you to

use Billawala's Pandora font family.

- `syntonly` will make \LaTeX only check the syntax of your document, while `tracefmt`, with its various options `errorshow`, `warningshow`, `infoshow`, and `debugshow`, allows you to trace NFSS as \LaTeX processes your document.
- `varioref` provides a way to automatically adapt the text of a reference, depending on the position of the `\label`.

Many other packages on CTAN already work with $\LaTeX 2_\epsilon$ or will soon be converted. In the first category one finds `a4`, `epsfig`, `exams`, `labels`, `layout`, the NTG document class family `artikel1`, `rapport3`, etc., `subeqnarray`, `psnfss`, `textfit`, while the latter contains the `babel` collection, `changebar`, `ltugboat` and friends, the 'Mainz' packages `array`, `ftnright`, `multicol`, `theorem`, `verbatim`, and `supertabular`.

8 Miscellaneous goodies

This section describes some features which are perhaps not used every day, but which can come in handy for solving certain practical document preparation problems.

8.1 Controlling page breaks

Sometimes, when preparing the *final* version of your document, you might need to help \LaTeX break the pages in a suitable way. \LaTeX 2.09 had commands like `\clearpage`, `\samepage`, etc., while $\LaTeX 2_\epsilon$ provides, in addition, commands which increase or decrease the height of the *current* page from its 'natural' height `\textheight` by an amount *size*.

```
\enlargethispage{size}
\enlargethispage*{size}
```

For example, `\enlargethispage{-\baselineskip}` decreases the length of the current page by one line, while `\enlargethispage*{2\baselineskip}` makes it two lines longer than usual.

The starred form also shrinks any vertical white space on the page as much as possible, so as to fit the maximum amount of text onto the page.

8.2 Floats

A new command and a new 'float specifier' will allow you more control over \LaTeX 's float placement algorithm.

```
\suppressfloats[placement]
```

This command stops any further floating environments from being placed on the current page. The optional

argument *placement* can be either `t` or `b` (not both), and in this case the restriction applies only to putting further floats at the top or at the bottom.

```
Extra float placement specifier: !
```

This can be used, along with at least one of `h`, `t`, `b` and `p`, in the float placement optional argument.

If a `!` is present then, just for this particular float, whenever it is processed by the float mechanism the following are ignored:

- all restrictions on the number of floats which can appear;
- all explicit restrictions on the amount of space on a text page which may be occupied by floats or must be occupied by text.

The mechanism will, however, still attempt to ensure that pages are not overfull and that floats of the same type are printed in the correct order.

Note that its presence has no effect on the production of float pages.

A `!` placement specifier overrides the effect of any `\suppressfloats` command for this particular float.

References

- [1] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The \LaTeX Companion*. Addison-Wesley, Reading, USA, 1994.
- [2] Leslie Lamport. *\LaTeX —A Document Preparation System. Second edition* Addison-Wesley, Reading, USA, 1994.
- [3] Frank Mittelbach and Rainer Schöpf. Towards \LaTeX 2.10. *TUGBoat*, 10(3):400–401, November 1990.
- [4] Frank Mittelbach. E- \TeX : Guidelines for future \TeX extensions. *TUGBoat*, 11(3):337–345, September 1990.
- [5] Frank Mittelbach and Chris Rowley", \LaTeX 2.09 \leftrightarrow \LaTeX 3. *TUGBoat*, 13(1):96–101, April 1992.
- [6] Chris Rowley. \LaTeX 3 update. *TUGBoat*, 13(3):390–391, October 1992.
- [7] Frank Mittelbach and Chris Rowley and Michael Downes. Volunteer work for the \LaTeX 3 project. *TUGBoat*, 13(4):510–515, December 1992.
- [8] Frank Mittelbach and Chris Rowley. Volunteer work for the \LaTeX 3 project. *\TeX and TUG NEWS*, 3(1):7–11, January 1994.

Standard Document Classes and Packages

A Cook's Tour of what is available in the basic system

Johannes Braams

PTT Dr Neher Laboratorium
Leidschendam
The Netherlands
j.l.braams@research.ptt.nl

Abstract

This article describes the Document Classes and Packages that are available or will be available shortly for use with $\text{\LaTeX}2_{\epsilon}$. The number of available Packages and Classes will obviously change once $\text{\LaTeX}2_{\epsilon}$ is the 'official' \LaTeX .

1 Filenames

When we¹ decided that we wanted to produce an intermediate version of \LaTeX after all, we also thought that it was a good idea to introduce a number of new file extensions to the \LaTeX world. The reason for this choice was to make it easy to distinguish between files that contain a Document Class, files that contain an external option to a Document Class and files that contain Packages. In table 1 you can find an overview of the extensions we have introduced. We would welcome it if you would stick to the same set of extensions when you upgrade your old `.sty` files.

2 What is available?

In the following sections an overview is given of the Document Classes and Packages that are currently available or that will become available shortly. First we focus on the files that are (will be) part of the standard $\text{\LaTeX}2_{\epsilon}$ distribution.

2.1 Document Classes in the $\text{\LaTeX}2_{\epsilon}$ distribution

In table 2 an overview is given of the Document Classes that are available when you get the standard distribution of $\text{\LaTeX}2_{\epsilon}$.

Most of these will be familiar to you, they are the successors of their \LaTeX 2.09 counterparts. Basically these Document Classes behave like the old document styles. But there are a few changes:

- The options `openbib` and `twocolumn` are now internal options, the files `openbib.sty` and `twocolumn.sty` do not exist any more.
- A number of new options are implemented; supporting a range of paper sizes. Currently implemented are `a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper` and `executivepaper`. The Document Class `proc` does not support `a5paper`. Besides these `papersize` options there is also the option `landscape` (which is not supported by the `letter` Document Class).
- The Document Class `slide` can now be used with \LaTeX , \SLiTeX does not exist as a separate format any

longer.

Two column slides are not supported.

When \LaTeX finds the file `sfonts.cfg` it will load it. This file can be used to customize the fonts used for making slides.

- The former `option proc.sty` has now been turned into a separate Document Class, which is implemented by building on `article` using the `\LoadClass` command. This Class does not allow the options `a5paper`, `b5paper` and `onecolumn`.

The Document Class `ltxdoc` is new. It is used in the documentation of all the $\text{\LaTeX}2_{\epsilon}$ source code. The Document Class is built upon the `article` class and also loads the `doc` package.

It defines the command `\DocInclude` which works like the `\include` command from \LaTeX , but setting things up for formatting documented source code.

The formatting of the source code can be customized by creating the file `ltxdoc.cfg`. Such a file could for instance select your favorite paper size. This can be done by entering the following command in `ltxdoc.cfg`:

```
\PassOptionsToClass{a4paper}{article}
```

Selecting `a5paper` is not allowed; the source listings wouldn't fit.

Another command defined by this Document Class is `\GetFileInfo`. It tries to retrieve the date, version and description information from the optional argument to any `\Provides...` command.

¹ 'We' are the $\text{\LaTeX}3$ project Team

2.2 Packages in the $\text{\LaTeX}2_{\epsilon}$ distribution

Table 3 lists the packages that are contained in the $\text{\LaTeX}2_{\epsilon}$ distribution.

All the packages in table 3, except `shortvrb` are described in ‘The \LaTeX Companion’².

The `ifthen` (which used to be `ifthen.sty` has been enhanced and now also defines `\newboolean`, `\setboolean` and `\boolean{...}` to provide a \LaTeX interface to \TeX ’s switches. Other new commands are `\lengthtest` and `\ifodd`.

The package `shortvrb` has only recently been introduced. It contains the definitions of the commands `\MakeShortVerb` and `\DeleteShortVerb` from the `doc` package. By providing this package those commands can also be used in other documents besides \LaTeX source

code documentation.

The `varioref` package wasn’t distributed with earlier versions of \LaTeX , unlike most of the other packages, which all were part of either the \LaTeX distribution or the `doc` and `NFSS` packages.

2.3 Other Classes and Packages

Table 4 lists the Packages and Document Classes that are available on CTAN at the time of the writing of this article.

The Document Classes from the NTG family are compatible with the standard Document Classes, but provide a different layout. They also provide a few extra options and a possibility to easily change the fonts used for headings.

In table 5 a few packages are listed of which the author is aware that they will appear on CTAN within the next months.

<code>dtx</code>	Documented source code
<code>fdd</code>	Documented source code for <code>.fd</code> files
<code>fd</code>	A font definition file
<code>ins</code>	DOCSTRIP instructions to unpack a <code>.dtx</code> file
<code>cls</code>	A stripped file containing a Document Class
<code>clo</code>	A stripped file containing an external option for a Document Class
<code>cfg</code>	An optional file that is looked for at runtime and which can contain customization code
<code>sty</code>	A stripped file that contains a (part of) a Package
<code>ltx</code>	A stripped file used when building the $\text{\LaTeX}2_{\epsilon}$ format
<code>def</code>	A stripped file containing definitions that will be read in at runtime.

Table 1: Extensions for $\text{\LaTeX}2_{\epsilon}$ files

<code>article</code>	produced from <code>classes.dtx</code>
<code>report</code>	produced from <code>classes.dtx</code>
<code>book</code>	produced from <code>classes.dtx</code>
<code>letter</code>	produced from <code>letter.dtx</code>
<code>slides</code>	produced from <code>slides.dtx</code>
<code>proc</code>	produced from <code>proc.dtx</code>
<code>ltxdoc</code>	produced from <code>ltxdoc.dtx</code>

Table 2: Available document classes that are part of $\text{\LaTeX}2_{\epsilon}$

²by Goossens, Mittelbach and Samarin; published by Addison-Wesley, ISBN 0–201–54199–8

ifthen	produced from ifthen.dtx
makeidx	produced from makeindx.dtx
showidx	produced from makeindx.dtx
doc	produced from doc.dtx
shortvrb	produced from doc.dtx
newlfont	produced from newlfont.dtx
oldlfont	produced from oldlfont.dtx
latexsym	produced from .dtx
exscale	produced from .dtx
eufrak	produced from eufrak.dtx
euscript	produced from euscript.dtx
oldgerm	produced from oldgerm.dtx
pandora	produced from pandora.dtx
syntonly	produced from syntonly.dtx
tracefnt	produced from tracefnt.dtx
varioref	produced from varioref.dtx

Table 3: Available document packages that are part of $\text{\LaTeX}2\epsilon$

a4	for typesetting on A4 pages, provides the option <code>widemargins</code>
artikel1	NTG family Document Class
artikel2	NTG family Document Class
artikel3	NTG family Document Class
boek	NTG family Document Class
boek3	NTG family Document Class
epsfig	a combination of <code>espf.sty</code> and <code>psfig.sty</code>
exams	for typesetting examination papers with questions and answers
labels	for printing labels
layout	a complete rewrite of Kent McPhersons <code>layout.sty</code>
rapport1	NTG family Document Class
rapport3	NTG family Document Class
subeqnarray	allows ‘sub-numbering’ for the <code>eqnarray</code> environment
psnfss	makes lots of PostScript fonts available to \LaTeX
textfit	fits text in a certain amount of space

Table 4: Packages and Document Classes available for $\text{\LaTeX}2\epsilon$ on CTAN on march 19, 1994.

array	Reimplementation of the <code>array</code> environment
babel	Support for many languages
brief	NTG family Document Class
changebar	Typesets change bars using <code>\special</code> commands
ftnright	Prints the footnotes of a page in the rightmost column
ltugboat	Document Class for TUGboat
multicol	Typeset in many columns
supertabular	For <code>tabular</code> environments that need many pages
theorem	A reimplementation of the <code>theorem</code> environment
verbatim	A reimplementation of the <code>verbatim</code> environment

Table 5: Packages and Document Classes to be made available for $\text{\LaTeX}2\epsilon$

Upgrading old styles

Johannes Braams

PTT Dr Neher Laboratorium
Leidschendam
The Netherlands
j.l.braams@research.ptt.nl

Abstract

This article deals with the things you have to look for when upgrading an old `.sty` file for $\text{\LaTeX}2_{\epsilon}$, turning it into either a Document Class or a Package.

1 Introduction

This article is meant for those \LaTeX users that have developed their own extensions to the \LaTeX system and want to retain that functionality when they start to use $\text{\LaTeX}2_{\epsilon}$.

2 A `.sty` file is not a `.sty` file

With the old \LaTeX 2.09, *all* extensions to the \LaTeX system were stored in files with the extension `.sty`. Therefore it was hard to tell without looking at the contents of a file whether it was a documentstyle, an option or a package implementing entirely new functionality. As you can see in table 1 in the article on Document Classes and Packages, we have changed that and introduced a number of new extensions.

This means that you will have to ask yourself when you upgrade an existing `.sty` file what kind of functionality it implements.

Here are a few points which might help you in deciding what to do with your `.sty` file.

- Was the original `.sty` file a documentstyle? Then turn it into a Document Class.
- Was the original `.sty` file meant to be used for a certain type of document? In that case you should consider turning it into a Document Class, possibly by building on top of an existing class. An example of this is `proc.sty` which is now `proc.cls`.
- Was it just changing some aspects of the way \LaTeX does things? In that case you would probably want to turn your `.sty` file into a Package.
- Was it adding completely new functionality to \LaTeX ? Examples of this kind of `.sty` file are files such as `seminar.sty` and `XYpic.sty`. This you most certainly will want to turn into a Package for $\text{\LaTeX}2_{\epsilon}$.

3 Getting it to work again

3.1 General

When you start to work on your `.sty` file the first thing to do is see if it runs with $\text{\LaTeX}2_{\epsilon}$ unmodified. This assumes that you have a suitable test set that tests all functionality

provided by the `.sty` file. (If you haven't, now is the time to make one!) We expect that quite a large number of the available `.sty` files will run with $\text{\LaTeX}2_{\epsilon}$ without any modification. Yet if it does run, please enter a note into the file that you have checked that it runs and resubmit it to the archives if it was a distributed file.

When your file `.sty` file uses commands that used to be part of the way \LaTeX used to deal with fonts than your file will almost certainly *not* work. You will have to look in the \LaTeX companion and the file `features.tex` which is part of the $\text{\LaTeX}2_{\epsilon}$ distribution to see what needs to be done. Macros such as `\tenrm` or `\xipt` do not exist any longer.

Also when you use internal commands from NFSS version 1 you will have to be very careful to check if everything still works as it was once intended.

When you need to debug your \TeX code you have to know that $\text{\LaTeX}2_{\epsilon}$ sets `\errorcontextlines` to `-1`. When you need more information from \TeX on where what kind of error occurs you most probably will want to change the value of `\errorcontextlines`.

3.2 Document styles

When upgrading your own document style you should start by asking yourself if it isn't more easy to reimplement it by building on an existing Document Class. This has now become very easy, you can pass any user supplied options to the underlying class with:

```
\DeclareOption*{%
  \PassOptionsToClass{%
    \CurrentOption}{class}}
```

You can make sure that certain options are used by default with:

```
\PassOptionsToClass{%
  option,option,...}{class}
```

You can disallow certain options with:

```
\DeclareOption{option}%
{\@latexerr{%
  Option 'option' not supported}}}
```

Then don't forget to call `\ProcessOptions` and `\LoadClass{class}`.

When you are upgrading a documentstyle, please note that some parameters that were in the \LaTeX 2.09 documentstyles have been removed. The parameter `\footheight`, which was given a value in some documentstyles, was never even used, but it took one of the precious dimension registers, so we decided to remove it. The change in the way floats are handled rendered `\@maxsep` and `\@dblmaxsep` useless. Removing these freed another two dimension registers.

Another subject to address when you are upgrading a documentstyle, is that of options. The way options are handled in $\text{\LaTeX}2_{\epsilon}$ is considerably different from \LaTeX 2.09. You now have to declare each option that a Document Class is to recognise. But you can now also give a Package a number of options, something that was impossible in \LaTeX 2.09.

When your Document Class doesn't support floating objects it is no longer necessary to set the parameters for the handling of floats. For instance, the Document Class `letter`, which doesn't support figures or tables no longer sets parameters such as `\floatsep`, `\textfloatsep`, etc.

When your documentsytle used to perform major surgery on the inner workings of \LaTeX , be very carefull when upgraing. A lot of code might have changed, so please check if your redefinitions are still valid. When you needed to redefin either `\begin{document}` or `\end{document}` to accomplish your wishes, please not that this might not be necessary any longer. We have introduced 'hooks' into these commands. These hooks can be accessed with `\AtBeginDocument` or `\AtEndDocument`.

4 Using $\text{\LaTeX}2_{\epsilon}$ features

When you upgrade your file (or write a new one) you may wish to use one or more of the new features of $\text{\LaTeX}2_{\epsilon}$. If you do, *please* include a line such as the following at the top of your file:

```
\NeedsTeXFormat{LaTeX2e}[1994/02/14]
```

This would give generate a warning if someone used your file with the first test release of $\text{\LaTeX}2_{\epsilon}$. When you want this mechanism to work for your file as well, be sure to use one of `\ProvidesClass`, `\ProvidesPackage` or `\ProvidesFile`.

The first argument to these commands is the name of your Document Class or Package³ and should match the name of the file it is stored in. The second argument is optional and should provide version information. Here is an example form the `a4` Package:

```
\ProvidesPackage{a4}%
[\filedate\space
```

³The first argument of `\ProvidesFile` should be a full filename, including the extension.

⁴A dirty trick for this would be to enclose new features in a test that checks whether `\documentclass` is defined

```
\fileversion\space
A4 based page layout]
```

Note that the macros `\filedate` and `\fileversion` were defined earlier in the Package file.

The session 'Advanced class(es) and packages' will discuss this subject in greater detail.

5 Compatibility with \LaTeX 2.09

The subject of remaining compatible with \LaTeX 2.09 is very complex. To start with you may want to remain fully compatible with the old version of \LaTeX because you work together with people that can't upgrade for some reason. This means that you will either be forced not to use *any* of the new features of $\text{\LaTeX}2_{\epsilon}$, or you will have to find a different solution⁴. Such a solution could be the use of `DOCSTRIP` modules to produce two different files from your code, one for the old \LaTeX installation and one to be used with $\text{\LaTeX}2_{\epsilon}$.

Another kind of compatibility is that which is provided in the compatibility mode of $\text{\LaTeX}2_{\epsilon}$. This mode was introduced to be able to run old \LaTeX 2.09 documents through $\text{\LaTeX}2_{\epsilon}$, yeilding (almost) the same result. If this is what you need to acheive than you may be please to know that the `\ifcompatibility` switch can be used to test for compatibility mode. Using this switch, you can develop a full blown $\text{\LaTeX}2_{\epsilon}$ Package or Document Class out of a \LaTeX 2.09 style file and yet still be able to print your old documents without changing them.

6 Possible Pitfalls

Some mistakes that might be easilly made and that can lead to unexpected results:

- You declare options in your Package using `\DeclareOption` but forget to call `\ProcessOptions`. $\text{\LaTeX}2_{\epsilon}$ will die horribly this way.
- The usage either of `\footheight`, `\@maxsep` or `\@dblmaxsep` outside of compatibility mode will lead to a complaint from \TeX about an unknown command sequence.
- With \LaTeX 2.09 the order in which options to a documentstyle were specified was *very* significant. A document might fail if the options are given in the wrong order. $\text{\LaTeX}2_{\epsilon}$ does not necessarily process the options in the order that they were specified. When the order is relvenat to your code you have to use `\ProcessOptions*` to let $\text{\LaTeX}2_{\epsilon}$ evaluate them in the order specified. For the `babel` package for instance, the order is significant, because the language specified last will be the one the document starts off with.

7 General advice

This section contains some general points of advice for people who are upgrading their `.sty` files.

- Carefully read the `features.tex` which is part of the $\text{\LaTeX}2_{\epsilon}$ distribution.
- When you distribute your work *please* distribute it in `.dtx` format and supply a `.ins` file to produce the stripped version(s) of your code.
- Use \LaTeX commands as much as possible; this will be of benefit to you when \LaTeX 3 appears.

7.1 Option naming conventions

No option names are built into the \LaTeX core system, but we hope that package and class writers will follow these conventions for names of options that have general applicability. Of course, packages may also have options with names particular to the package.

language options If your package produces ‘fixed strings’ that produce text in the output then please

consider supporting language options as defined by the `babel` package.

paper size options The standard classes support the options `letterpaper`, `legalpaper`, `executivepaper`, `a4paper`, `a5paper`, `b5paper`; any paper size can be modified by the option `landscape`. These set the `\paperheight` and `\paperwidth` parameters. When writing a class that supports different paper sizes you should make similar options, with names ending in `paper`, for those sizes that you support.

driver options Many graphics related packages need to know what dvi-driver is being used (so that the correct syntax may be used in the `\special` command). It is hoped that conventional option-names will be agreed denoting the major drivers. Currently the suggested list includes: `emtex`, `dvips`, `oztex` etc.

debugging options `errorshow`, `warningshow`, `info` `show`, `debugshow`, `pausing`.

A brief guide to T_EX assistants*

Sebastian Rahtz

ArchaeoInformatica
York, England
spqr@ftp.tex.ac.uk

This paper offers a ‘back to basics’ overview of the various types of software which are of assistance to the T_EXnical writer on a personal computer (this includes Unix machines, but excludes, for instance, VMS or VM/CMS systems). It is based on a presentation to the October UK-TUG meeting, and I am grateful to the other speakers, and our chairman Allan Reese, for insights and news. I would also like to refer readers to the Dutch T_EX Group’s journal *MAPS*, whose issue 93.2 contains a variety of useful papers on T_EX interfaces (two of which are reproduced in Baskerville Volume 3.2 (December 1993)).

Every T_EX user knows that the traditional command-line way of working (the ‘edit ; compile; {preview, print}’ cycle) is far from ideal, and many people want some help. We may define eight different ways in which the environment we interact with can be improved:

1. Integrated environments (e.g. T_EXtures on Mac), where everything is tightly-coupled, and designed from the ground up to work together;
2. Integrating shells (e.g. T_EXshell), where a set of different programs are controlled by a single interface, which does its best to protect the user from their idiosyncrasies and errors;
3. Loosely-coupled packages (e.g. Windows, Desqview setups), where a multi-tasking environment is used to combine together various programs to provide a useful working environment;
4. T_EX-aware editors (e.g. Gnuemacs AUCT_EX package), which simplify the input of complex markup, and check syntax;
5. T_EX super-scripts, including Makefile generators (e.g. ‘textit’); these take away the need to remember complicated program switches, and remembering which files are up to date, by understanding the rules about the different types of T_EX-related files;
6. Super-previewers (e.g. xtex), which allow us to make greater use of the previewer to inspect and interact with our document.
7. Hypertext source convertors (e.g. latex2html), which make the tedium of writing generic markup with L^AT_EX more worthwhile by giving it a life beyond the printed page;
8. Friendly T_EX programming; when we *do* interact with T_EX’s enigmatic ‘*’ prompt, we want macros to help

us, not hinder us. This is discussed by Jonathan Fine in Baskerville Volume 3.2 (December 1993).

Since most computer users seem to belong in distinct ‘camps’, based on machines and operating systems, it seems appropriate to examine what is on offer in each area.

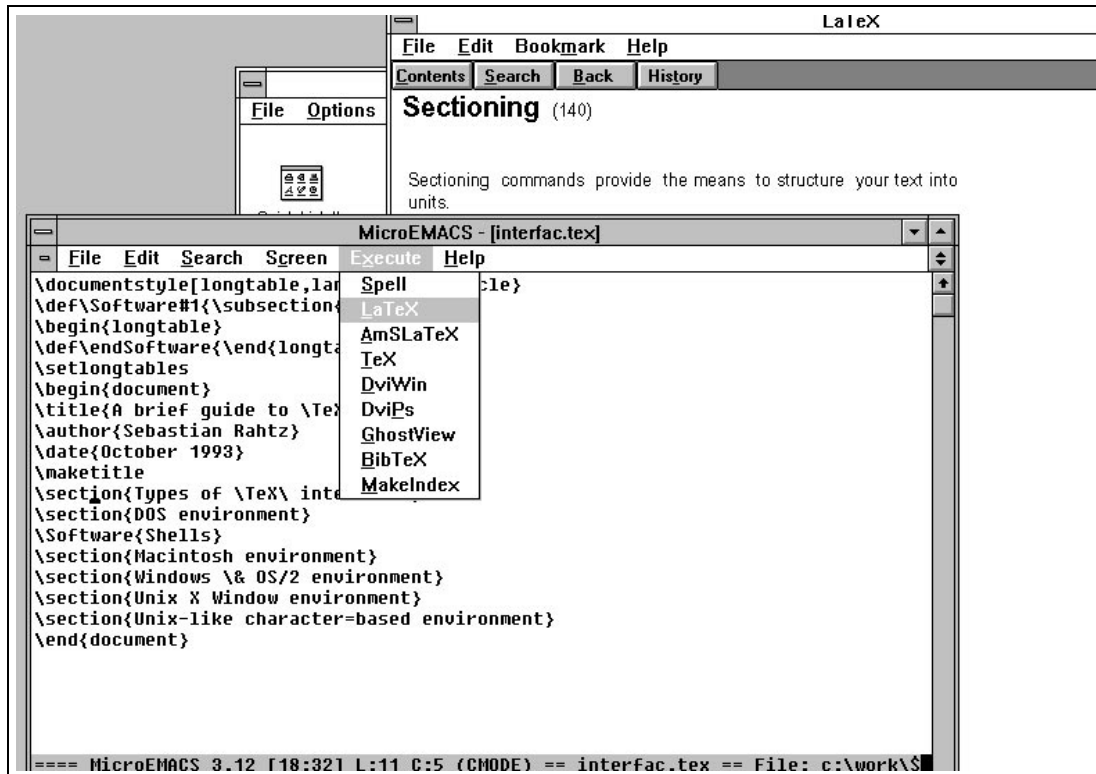
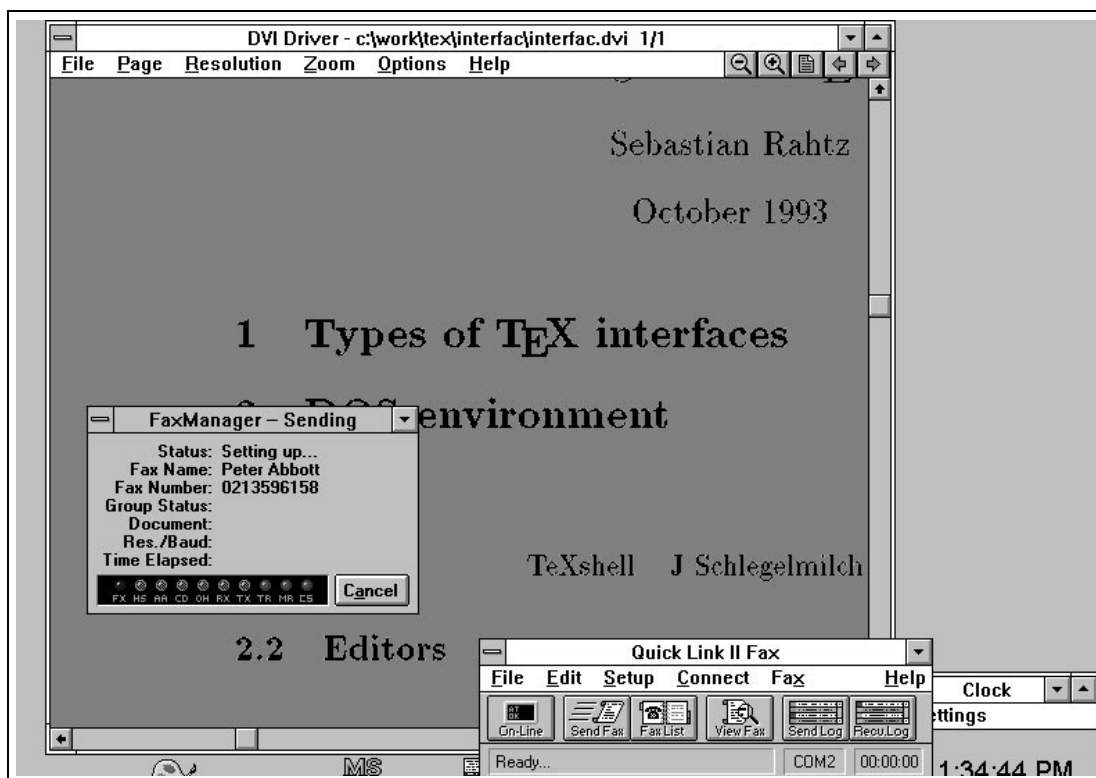
1 Windows, OS/2, and NT

For a fully-fledged Windows T_EX, we have to look to the commercial companies: TurboT_EX, from Kinch Computer Co, offers a full T_EX, previewer, and drivers with a moderately-well integrated Windows interface, although (critically) it lacks a real integrated editor. The ‘traditional’ PC supplier of T_EX, PCT_EX, has also recently released a Windows setup, although it (like TurboT_EX) fails to take advantage of builtin fonts properly. Y&Y sell a high-quality Windows previewer and PostScript driver, linked with a public domain T_EX and commercial programmer’s editor, which work together well in a loosely-coupled way, and offer genuine integration with the Windows font systems (although the previewer has trouble using PK fonts). *Scientific Word* takes a different direction by offering a WYSIWYG writers interface (particularly for maths) which saves files in L^AT_EX format and does final formatting for printing with TurboT_EX. None of these products, however, offer the same degree of integration and instant previewing as Textures on a Mac, and we still await the ‘real’ Windows T_EX.

The more impoverished writer who wants to build her own Windows T_EX setup has a not altogether easy time.¹ There is an excellent previewer, Hippocrates Sendoukas’ *Dviwin* (now at version 2.81, but *still* lacking support for virtual fonts, and nowhere near using the Windows system fonts), and the excellent emT_EX ports of the basic T_EX programs run in DOS windows (with some shenanigans necessary to change extended memory managers), but integration is left to the user. The powerful editor Microemacs has an adequate Windows port, extensible to allow T_EX jobs (compiling, previewing, printing — see Figure 1) and context-sensitive help to be selected from menus, but it does not provide much protection for the user from the essentially unsatisfactory nature of Windows DOS boxes and multi-

*Reprint from the Annals of the UK T_EX Users’ Group: **Baskerville**, Volume 3.2, December 1993.

¹Wietse Dol discusses this more fully in Baskerville Volume 3.2 (December 1993).

Figure 1: *Microemacs* \TeX menusFigure 2: *Dviwin* accessing Windows printer driver

tasking. The alternative, running a DOS \TeX shell in a DOS box under Windows, probably gives a better result (see Wietse Dol's article). But the convenience of access to Windows printer drivers (such as fax boards — see Fig-

ure 2) is a pointer to why this possibility should not be ignored.

Those converting to NT will, of course, have access to all the Windows tools, and the Unix ‘web2c’ \TeX has been ported, but there do not yet appear to be any interfaces which make serious use of NT’s abilities.

OS/2 users are well served by the basic $\text{em}\TeX$ kit, which has a proper Presentation Manager previewer, and by various shells (e.g. $\text{PM}\TeX$) which make life easy. Given the good multi-tasking, Windows compatibility, and the availability of compilers to port most Unix tools (such as *Gnuemacs*, *make* etc), the OS/2 \TeX ie has the best of almost all worlds.

2 DOS

The hard-pressed DOS user (or lover — there are a few) has a plethora of tools to make up for the inadequacies of the command line. The most mature of the ‘shell’ programs, which run \TeX , previewer, printer, editor etc for you sequentially when you select menu items from a character-based screen, is probably Jürgen Schlegelmilch’s \TeX shell (see Phons Bloemen’s article below). A typical screen is shown in Figure 3. Other offerings include Thomas Esken’s \TeX surface, Johannes Martin’s \TeX pert, Ulrich Jahnz’s $\text{Eddi4}\TeX$, the editors Redit and ET (see Reese’s article above) and Petr Olsak’s menu builder ‘mnu’. All of these have some or all of the following characteristics:

- a model of the original Turbo Pascal — character screens with pull-down menu bars;
- protection from the horrors of DOS environment variables controlling programs from four different authors (*after* you have got the shell properly configured);
- \TeX and other programs activated by ‘shelling out’ to DOS, (an intrinsically ‘uncool’ procedure), with parsing of error messages;
- heavy use of customizable environments (see Figure 4), which can confuse the novice to screaming point;
- limited editors which can be difficult to use with ‘industrial strength’ documents;
- access to online \TeX context-sensitive help.

This is not to deny the great usefulness of parts of all of the DOS \TeX shells, but their parts are greater than their sum. The dedicated DOS \TeX user will find them extremely useful, but perhaps they can be summed up by saying that they will not convert many Microsoft Word users.

A recent initiative by Dutch \TeX users has resulted in $\text{4}\TeX$ (see the article by Wietse Dol), a \TeX shell accompanied by a packaging of almost all the good public domain \TeX goodies. This is undoubtedly the system which is getting most support and development. The even more powerful $\text{As}\TeX$ setup of Michel Lavaud (described in *TUGboat* 14.3, 238–44) unfortunately depends on commercial software for its implementation, but those interested in a wider perspective on *managing* their \TeX document library should examine this closely.

Those who want the approach of Scientific Word (WYSIWYG maths composition) in plain DOS can try the public

domain ET by John Collins, or the commercial *Leo* from ABK Software.

3 Macintosh systems

The Macintosh user does not need convincing about GUIs or integration; for hard cash, she can buy what is probably the fastest (given the processor), slickest \TeX around, Textures. A port of the Unix \TeX is available (C $\text{Mac}\TeX$), and a half-way house, the shareware $\text{Oz}\TeX$, which offers integration between a good compiler, previewer and printer driver, but not quite the smooth transition of Textures, or the use of system fonts. What is lacking on a Mac is a WYSIWYG interface to mathematical writing such as that offered by Scientific Word.

4 Unix, etc

The Unix \TeX user has a rich variety of tools to play with; like everything else in Unix, they are dangerously powerful, and require dedication to obtain true mastery. They start with simple things like the *cs*h or *ba*sh history mechanisms, which let you recall previous commands by name:

```
$ history
 171  lutex fine2
 172  dvips fine2
 175  latex interfac
 176  lpq
 180  dvips fine2 -o fine2.ps
 182  cp fine2.ps /user
$ !lat
$ !dv
```

Another universal Unix tool is ‘make’ which manages the relationship between different stages of a document’s ‘compilation’ from various sources, and keeps things up to date. A large book, with dozens of input files, hundreds of graphics inclusions from different programs, a bibliography managed with $\text{Bib}\TeX$, and an index, is likely to need such a tool to keep the author sane. There are various tools to help you write ‘make’ files, and to do similar jobs to ‘make’:

imaketex Just write an *imake* file and you just type ‘make’; as in ‘just write’...

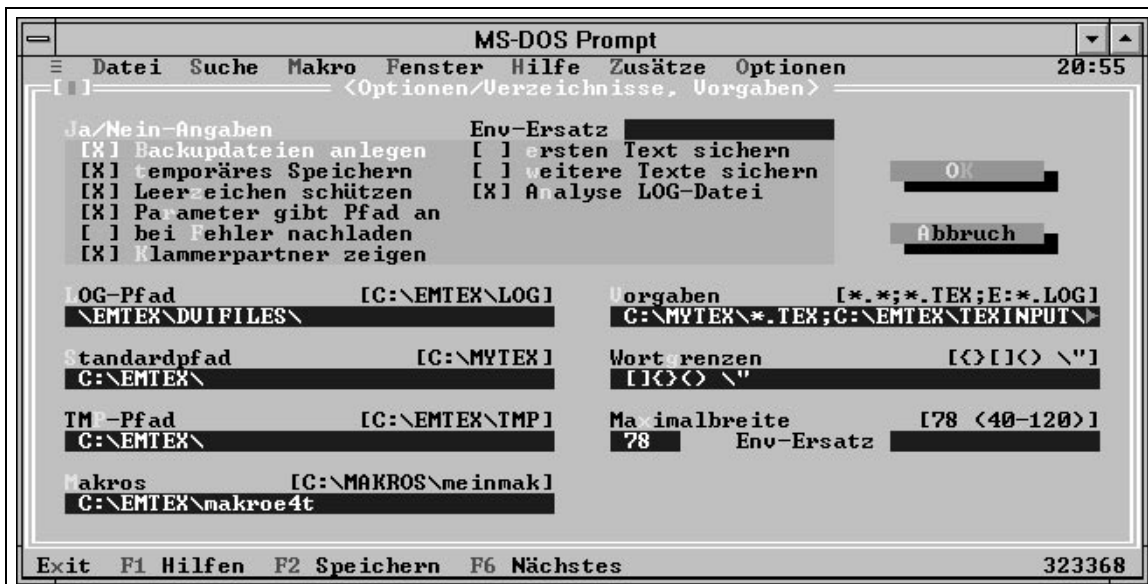
texit A Perl script which makes educated guesses about the state of your \TeX file and helps you decide what to do next;

latexmk Given a $\text{L}\TeX$ file this tries to work out what to do, like a ‘makefile’; but does not understand multiple-level dependencies;

latex.make A set of rules for real ‘make’, which understand a wide range of common \TeX jobs; but tailoring is not for the beginner. Sample ‘make’ rules are shown in Figure 5.

If you are already a Unix programmer, these are well worth investigating.

There do not appear to be any obvious public domain \TeX shells for the X Window system; the ‘doc’ editor does provide help in writing $\text{L}\TeX$, with limited immediate formatting, but it still leaves you to run \TeX . The Berkeley $\text{Vor}\TeX$ program to produce an interactive \TeX ended some years ago without a real success (although various results

Figure 3: \TeX shell, after running \TeX and finding an errorFigure 4: Eddi4 \TeX customization

have now been released for general use; they are available in CTAN).

The queen of Unix software is Gnuemacs; as well as being an *extremely* sophisticated editor, it provides hooks to run jobs synchronously or asynchronously from within the editor, and parse the output. The latest X Window version allows for user-defined menu bars, and multiple windows, which allows the Unix user access to as sophisticated a setup as almost everything except Textures. The actual editing can be enhanced with powerful Lisp routines to check syntax provide structure skeletons, edit in outline mode etc; the best-supported and enhanced package is Kresten Krab Thorup and Per Abrahamsen's AUCT \TeX . If you can learn to be happy with Gnuemacs, this has al-

most limitless possibilities. An example screen is given in Figure 6.

5 Conclusions

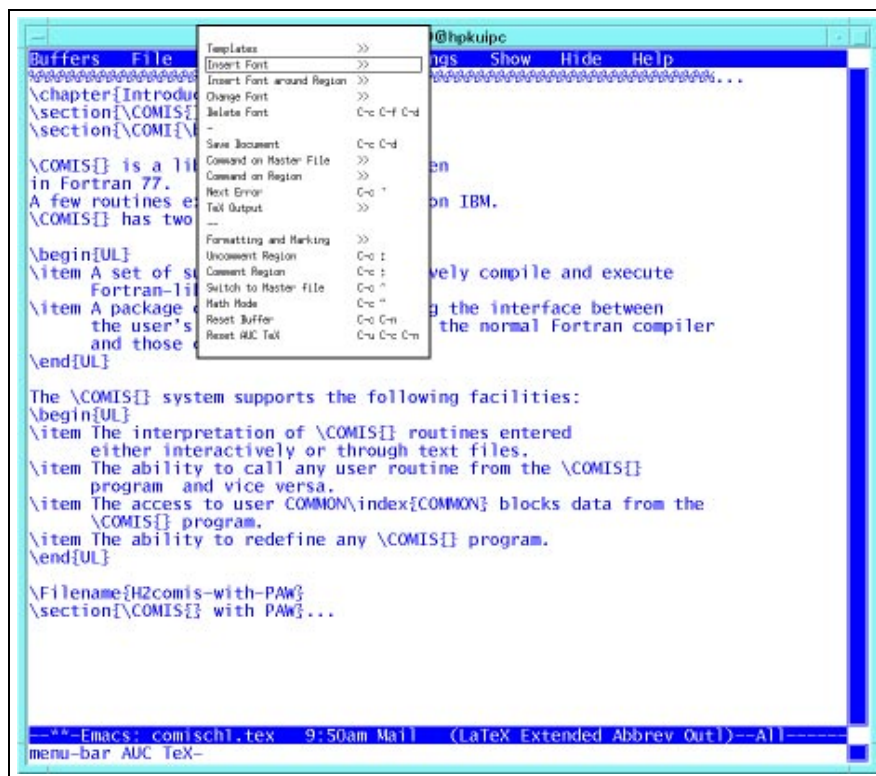
If you have to choose a \TeX environment, how do you start evaluating the offerings? I would suggest that base your choice on the following criteria:

- The editor; this is where you spend most of the time;
- On-line help; \LaTeX is unmemorable;
- Integration with windowing systems; doing just \TeX is dull;
- Lack of special features; compatibility with other \TeX setups is important;
- Integration with the 'style' of the environment (i.e. use of Adobe Type Manager).

```

# ensure that the bbl file gets regenerated if the bib file is changed
%.bbl : $(BIBFILES)
    @# if there is no aux file, skip this, it will get done later
    -@if [ -r $*.aux ] ;\
    then $(BIBTEX) $* ; \
    fi
# create a dvi file from a tex file
% %.dvi: %.tex
    $(LATEX) $*
    -@egrep -c 'Citation .* undefined.' $*.log && ($(BIBTEX) $*;$LATEX) $*)
    -@grep 'Rerun to get cross-references right' $*.log && $(LATEX) $*
# create postscript file of a dvi file
%.ps: %.dvi
    echo dvips -D $(PRINT_RES) -p $(PAGE_F) -l $(PAGE_T) -o $@ $*
    dvips -D $(PRINT_RES) -p $(PAGE_F) -l $(PAGE_T) -o $@ $*

```

Figure 5: An example of 'make' rules for a \TeX fileFigure 6: Gnuemacs and AUCT \TeX

For the five possible environments you might consider choosing:

- AUCT \TeX : if you use the X Window system and *gnuemacs*, there is no other way to live;
- \TeX shell: if you like plain DOS, this is well-designed and easily customized;
- Dviwindo: it really is a Windows application, and understands PostScript fonts;

- \TeX t \TeX ures: it works; it has semi-instant previewing; it integrates its fonts with the rest of the Mac;
- OS/2 and em \TeX : a very good command-line \TeX , a Presentation Manager previewer, and integration with a reliable operating system.

[Note: all the public domain or shareware software referred to in this article can be found in the CTAN archives. The commercial vendors all advertise in TUGboat.]

Building virtual fonts with 'fontinst'*

Alan Jeffrey

alanje@cogs.susx.ac.uk

Abstract

This document gives a brief overview of the `fontinst` package. The `fontinst` package is used to build *virtual fonts* (VFs) which allow PostScript fonts to be used as drop-in replacements for the Computer Modern fonts in \TeX .

Below, I'll describe VFs briefly, and describe how they can be built using the `fontinst` package.

1 A problem with fonts

One of the biggest problems about using fonts in \TeX is *encodings*, that is the order the characters come in the font. For example, the default encoding for Adobe's Times-Roman font is the 'Adobe Standard encoding':

```
! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~
ı ç £ / ¥ f § ¤ ' “ « ‹ › fi fl
- † ‡ † ~ - ¶ • : „ ” » ..%o
, , ^ ~ - ¶ • : „ ” » ..%o

```

```
Æ a      Ł Ø Æ °
æ      ı      ł ø œ ß
```

The default encoding for \TeX , however, is the ' \TeX text encoding'. The Adobe Times-Roman font in the ' \TeX text encoding' is:¹

```
Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω ff fi fl ffff
ı ■ ◻ ◂ ◃ ◅ ◆ ◇ ◈ ◉ ◊ ◌ ◍ ◎ ●
■ ! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; j = ç ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ “ ] ^ ·
` a b c d e f g h i j k l m n o
p q r s t u v w x y z — ~ ~ ..
      Ł
      ł
```

There are many other competing encodings: 'ISO Latin-1', ' \TeX extended text' (or 'Cork'), 'Macintosh', the list is seemingly endless.

In addition, different encodings contain different glyphs. The \TeX text encoding is supposed to contain a dotless 'j' character, and a slash for building 'ı' and 'Ł', but very few fonts contain these characters, and their places are taken by black squares above.

The problem of incompatible font encodings is addressed in \TeX by *virtual fonts*.

2 A solution: virtual fonts

As far as \TeX is concerned a virtual font (VF) is a font like any other. It has a \TeX font metric file, which contains the dimensions of each character, together with ligatures, kerning, and other typographical information.

However, a VF does not have an associated bitmap, Type 1 font, TrueType font, or other information about what the font should look like.

Instead, a VF has an associated `.vf` file, which contains a small fragment of DVI file for each character in the font. This DVI fragment may contain characters from other fonts, rules or `\specials`.

For example, the 'Adobe Standard' encoded Times-Roman font above is a 'raw' Type 1 font, but the ' \TeX text' encoded Times-Roman font is a virtual font.

- The 'ff', 'ffi' and 'fff' ligatures are faked by putting an 'f' next to an 'f', 'fi' or 'ff'.
- The missing 'dotless j' and 'ı-slash' are rules, together with a 'Warning: missing glyph' `\special`.
- The Greek upper case come from the Symbol Type 1 font.
- The other characters come from the Times-Roman Type 1 font.

Any DVI driver which understands VFs and can use Type 1 fonts can use the \TeX text Times-Roman VF as a drop-in replacement for Computer Modern.

*Reprint from the Annals of the UK \TeX Users' Group: **Baskerville**, Volume 4.1, Januari 1994.

¹The \TeX nically minded may note that the glyphs 'ı' and 'Ł' are not normally in the ' \TeX text encoding'. This is because Computer Modern has a special 'ı-slash' glyph for building 'ı' and 'Ł', which Adobe Times-Roman does not have. Its place is therefore taken by a black square, and there are ligatures with 'ı' and 'Ł' to produce 'ı' and 'Ł'. Thus this font is drop-in compatible with Computer Modern, despite the lack of an 'ı-slash' glyph.

3 A problem with virtual fonts

One stumbling block about using VFs is that they are not very easy to generate. Despite having been in existence for four years, there are very few tools for creating VFs.

The most important tool is Knuth’s `vptovf`, which converts *Virtual Property Lists* (vpls) into VFs. Unfortunately, the vpl language is rather opaque; for example the vpl code for the Adobe Times character ‘ff’ is:

```
(CHARACTER D 11
  (CHARWD R 6.47998)
  (CHARHT R 6.81995)
  (CHARDP R 0.0)
  (MAP
    (SELECTFONT D 1)
    (SETCHAR D 102)
    (MOVERIGHT R -0.17993)
    (SETCHAR D 102)
  )
)
```

Editing vpl files by hand is something of a black art, and there are few tools for manipulating them.

The main tool for generating vpls is Rokicki’s `afm2tfm`, which converts the *Adobe Font Metric (AFM)* files which come with every PostScript font into vpls. Unfortunately, `afm2tfm` cannot produce fonts with more than one raw font (for example the ‘ \TeX text’ encoded Times-Roman uses Symbol for the upper case Greek) and had problems with math fonts.

4 A solution: the ‘fontinst’ package

The `fontinst` package is designed to read AFMs and produce vpls. It:

- Is written in \TeX , for maximum portability (at the cost of speed).
- Supports the \TeX text, \TeX math, and extended \TeX text encoding.
- Allows fonts to be generated in an arbitrary encoding, with arbitrary ‘fake’ characters—for example the ‘ff’ character can be faked if necessary by putting an ‘f’ next to a ‘f’.
- Allows caps and small caps fonts with letter spacing and kerning.
- Allows kerning to be shared between characters; for example ‘ \acute{A} ’ can be kerned as if it were an ‘A’. This is

useful, since many PostScript fonts only include kerning information for characters without accents.

- Allows the generation of math fonts.
- Allows more than one PostScript font to contribute to a \TeX font; for example the ‘ff’ ligature can be taken from the Expert encoding, if you have it.
- Automatically generates an `fd` file for use with $\LaTeX_{2\epsilon}$.

The `fontinst` package is available as freeware from the CTAN archives, along with a selection of VFs which have been generated with `fontinst`.

Version 0.19 of `fontinst` is described in the proceedings of the Aston TUG AGM (*TUGboat* 14(3)). This description is now largely out of date.

The VFs generated by `fontinst` will be the standard VFs for use with Sebastian Rahtz’s `psnfss` package for $\LaTeX_{2\epsilon}$.

5 Using the ‘fontinst’ package

The `fontinst` package comes with full documentation in the file `fontinst.tex`. The simplest way to start to use `fontinst` is to edit the file `fonttime.tex`, shown in Table 1. This tells \TeX to create the Adobe Times Roman fonts in the ‘ \TeX extended text’ (T1) encoding, using the files:

- `ptmr0.afm`, `ptmri0.afm`, `ptmb0.afm` and `ptmbi0.afm`, the Times-Roman afm files.
- `latin.mtx`, the *TeX metric* file containing the default Latin characters.
- `T1.etx` and `T1c.etx`, the *TeX encoding* files containing the ‘ \TeX extended text’ and ‘ \TeX extended text caps & small caps’ encodings.

This produces a number of PL and vpl fonts, which can be converted into \TeX fonts using `pltotf` and `vptovf`.

For example by replacing every occurrence of `ptm` by `ppl` you can install the Adobe Palatino fonts.

If you generate any fonts with `fontinst` which you think other people might want to use, please send them to me, and if I like them, I’ll include them in the `fontinst` contributors directory.

```
\input fontinst.sty
\needsfontinstversion{1.303}

\installfonts
  \installfamily{T1}{ptm}{}
  \installfont{ptmrq}{ptmr0,latin}{T1}{T1}{ptm}{m}{n}{}
  \installfont{ptmrcq}{ptmr0,latin}{T1c}{T1}{ptm}{m}{sc}{}
  \installfont{ptmriq}{ptmri0,latin}{T1}{T1}{ptm}{m}{it}{}
  \installfont{ptmbq}{ptmb0,latin}{T1}{T1}{ptm}{bx}{n}{}
  \installfont{ptmbcq}{ptmb0,latin}{T1c}{T1}{ptm}{bx}{sc}{}
  \installfont{ptmbiq}{ptmbi0,latin}{T1}{T1}{ptm}{bx}{it}{}
\endinstallfonts
\bye
```

Table 1: The file `fonttime.tex`

Text to Hypertext Conversion with $\text{\LaTeX}2\text{HTML}$ *

Nikos Drakos

Computer Based Learning Unit,
University of Leeds,
Leeds LS2 9JT, UK.

email: nikos@cbl.leeds.ac.uk

www: <http://cbl.leeds.ac.uk/nikos/personal.html>

April 1, 1994

Abstract

$\text{\LaTeX}2\text{HTML}$ is a conversion tool that allows existing documents written in \LaTeX to become part of a global multimedia system. This paper presents some of the reasons for using such a system and describes the basic conversion process.

1 World Wide Web - A Global Multimedia System

Imagine a system that links all the text, data, digital sounds, graphics and video on all the world's computers into a single interlinked hypermedia 'web'. This is the potential of the Internet-based World Wide Web (WWW or W3) project ... [2]

The World Wide Web merges hypermedia techniques with networked document retrieval to provide a global information system of linked documents. These are traversed by 'clicking' in textual or iconic active areas, or searched via query mechanisms [1]. Hypertext links may point to a different location in the same document or to another document which may be located perhaps in another continent!

Documents are not limited to containing only textual information and may include high resolution images, audio and video samples. WWW also encompasses most of the services currently available on the Internet such as Usenet news, ftp, wais, archie, etc. Access to these services as well as the invocation of arbitrary computer programs (e.g. a database access or a simulation) is completely transparent to the user who sees them all as part of some document and interacts with them in a uniform and intuitive way.

Multimedia documents are written in a language designed specifically for the World Wide Web called HTML (HyperText Markup Language) which is based on SGML (Structured Generalised Markup Language). Documents are written by information providers who just place them on the WWW using a 'server' program. Then anyone with access to the Internet can use a 'client' or 'browser' program to access and view available documents. Clients and servers communicate via the HTTP protocol (HyperText Transfer Protocol). Apart from navigation facilities,

browsers also allow full text searches, 'cut and paste', text or audio annotations, personal 'hotlists', saving and printing in multiple formats and others. Such browser and server programs are freely available for most popular computer configurations.

With the explosive growth of the World Wide Web (500-fold since the first graphical browsers were made available this year [3]), and a potential audience of 15 million in more than 50 countries, providing information via the WWW is becoming an extremely attractive proposition.

2 \LaTeX to HTML Conversion: Why?

HTML is quite a simple markup language to learn and use. It allows basic formatting commands, bulleted lists, 'inlined' images, and hypertext links to other documents, multimedia sources, internet services or computer programs. But despite (and because of) its simplicity it has created a few headaches for information providers:

- there are no intuitive authoring tools (yet);
- yet another hypertext language has to be learned;
- existing documents available in other formats have to be reprocessed;
- hypertext document 'webs' are difficult to maintain;
- it is difficult or impossible to create highly formatted documents in HTML.

A flexible text to hypertext conversion tool can help in addressing these problems. The authoring problem simply disappears, existing documents can be reused immediately and a complex web of interlinked documents can be generated from a single source document. The automatic inclusion of formatted information such as tables or mathematical equations as inlined images also bypasses another serious problem with HTML. An additional benefit is that the paper-based version of a document can also be obtained from the same source.

*This is an updated version of a paper which appeared in Baskerville [4]

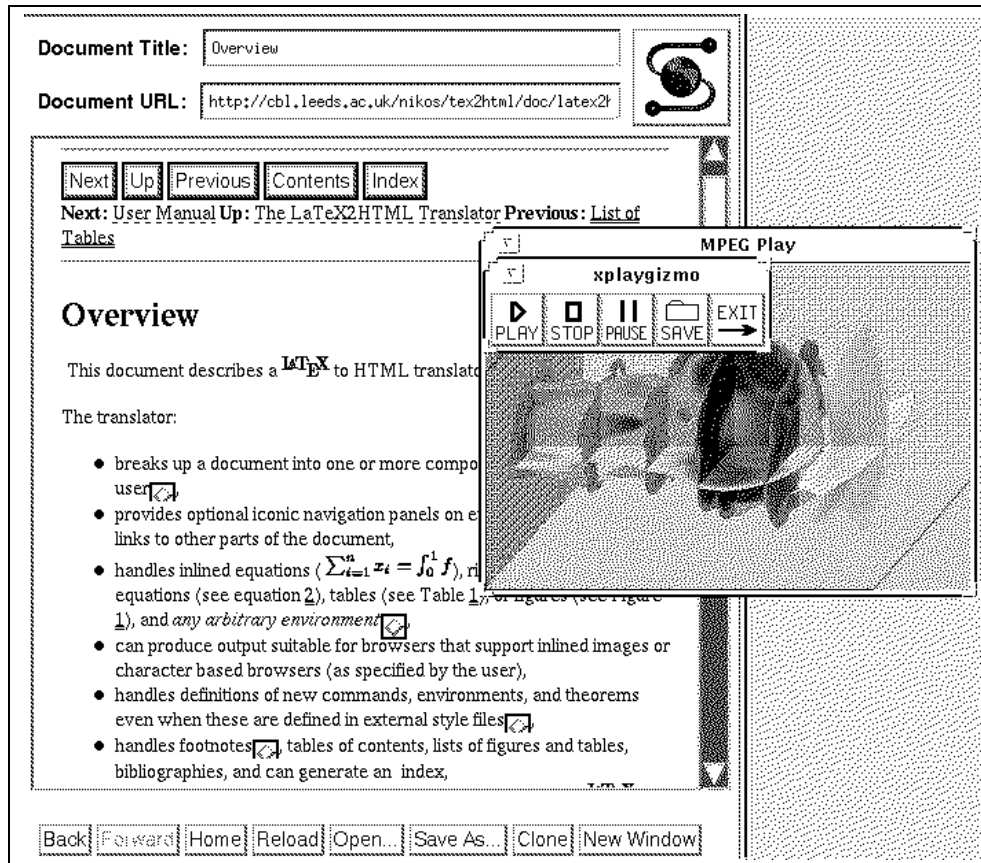


Figure 1: A converted document displayed using Mosaic with an external movie player triggered from it.

The utility of a conversion tool like L^AT_EX2HTML can be seen from the variety of contexts in which it has been applied. Some examples are listed below.

- Electronic books (such as that being produced by the Computational Science Education Project¹ sponsored by the US Department of Energy and involving 35 authors or the CRS4 Active Books Library² in Italy from which it is possible to interact with remote programs).
- Scientific papers such as those on the MIT Transit Project³ or this paper⁴!
- Lecture Notes, Supporting Documentation and Coursework⁵
- Online training material⁶
- General documents such as one recommending the use of L^AT_EX2HTML for the electronic submission of manuscripts to an IEEE journal⁷.
- System Documentation⁸ and User Manuals⁹.

3 L^AT_EX to HTML conversion: How?

The basic conversion process relies on the ability to distinguish between the *structure*, the *content* and the *formatting* information in a L^AT_EX document.

On the basis of sectioning information, a document is broken into separate parts and an iconic navigation mechanism is constructed in HTML which reflects this structure and allows a user to ‘jump’ between different parts. The cross-references, citations, footnotes, the table of contents and the lists of figures and tables are also translated into hypertext links. Formatting information which has equivalent ‘tags’ in HTML (lists, quotes, paragraph breaks, type styles, etc.) is also converted appropriately.

Although in most cases the loss of some formatting information (e.g. page margins or line widths) is harmless, there are occasions where the format has meaning e.g. when dealing with tables or user defined environments. Another problem is the replication of the mathematical equations

¹ <http://compsci.cas.vanderbilt.edu/csep.html>

² http://www.crs4.it/HTML/int_book/meta_page.html

³ <http://www.ai.mit.edu/projects/transit/tn-cat.html>

⁴ <http://cbl.leeds.ac.uk/nikos/doc/maps/maps.html>

⁵ http://www.cm.cf.ac.uk/lecture_notes.html

⁶ <http://www.strath.ac.uk/CC/Courses/OnlineTraining.html>

⁷ <http://www.research.att.com/esubmit/esubmit.html>

⁸ <http://www.cwi.nl/cwi/people/Guido.van.Rossum/python-tut/tut.html>

⁹ <http://olt.et.tudelft.nl/usr1/patrick/public.html/docs/wwman/wwman.html>

which must retain both their precise format as well as any of the predefined special mathematical symbols.

The innovative solution in such cases relies on the ability of HTML browsers to display inlined images inside the main text. Any part of a \LaTeX document for which it is not obvious how it should be translated directly into HTML is extracted from the main document and then placed on a pipeline (from latex to dvi, then postscript, then PPM and finally to GIF or XBM) which converts it into an image. Each image is then placed at the correct position in the final HTML document. Special care is taken to preserve contextual information that may affect the contents of each image (counter values, labels, references, active style files etc). An example of a converted document can be seen in Figure 1.

4 Hypermedia Extensions to \LaTeX

Apart from the obvious hypertext links within a \LaTeX document (e.g. navigation between sections, cross-references and citations) it is also possible to take full advantage of the HTML links to arbitrary multimedia sources (e.g. audio or video), electronic forms, and other remote documents or internet services.

This can be done with some new commands defined in a separate style file (`html.sty`) which are processed in a special way by the $\text{\LaTeX}2\text{HTML}$ translator. This style file defines commands for embedding external hypertext links, for extending the basic `\ref-\label` mechanism to operate between remote documents, and specifying that some text should only appear in the paper-based version or only in the HTML document. In most cases these commands have no effect when processed in the conventional way.

Another command allows the inclusion of arbitrary HTML markup directly in a \LaTeX document. This can be used to take advantage of new HTML facilities as soon as they become available (HTML is currently evolving towards a new specification called HTML+). A particularly good use of this feature is in the creation of interactive electronic forms from within a \LaTeX document.

5 Concluding Remarks

Conversion tools like $\text{\LaTeX}2\text{HTML}$ provide an easy migration path from familiar concepts towards authoring complex and format-rich hypermedia documents. In this way, familiarity with a system like \LaTeX makes it possible to contribute to and benefit from a rapidly expanding global hypermedia network.

References

- [1] T. Berners-Lee, R. Cailliau, J. Groff, and B. Pollerman. Worldwide web: The information universe. *Electronic Networking: Research, Application and Policy*, (1), 1992.
- [2] Joe Levy. The world in a web. *The Guardian*, page 19, November 11 1993.
- [3] Vern Paxson. Growth trends in wide-area TCP connections. *IEEE Network*, 1993. Available at <ftp://ftp.ee.lbl.gov/WAN-TCP-growth-trends.revised.ps.Z>.
- [4] Nikos Drakos. Text to Hypertext conversion with $\text{\LaTeX}2\text{HTML}$. Baskerville, December 1993.

A Further Information

$\text{\LaTeX}2\text{HTML}$ is written in Perl and requires freely available software. More information on how to get, install and use it is available via the WWW¹⁰ or using anonymous ftp from <ftp.tex.ac.uk> in `pub/archive/support/latex2html`.

Several computers on the Internet have public access World Wide Web clients accessible by telnet e.g.

- telnet info.cern.ch (direct connection - no username or password required)
- telnet ukanaix.cc.ukans.edu ('Lynx' requires a vt100 terminal. Log in as www.)

Information on the WWW is also available via anonymous ftp from <ftp.germany.eu.net> in `pub/infosystems/www`.

The Mosaic clients are in the directory `/pub/infosystems/www/ncsa/Web`.

¹⁰<http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html>

Colour slides with L^AT_EX and seminar.sty ¹

Michel Goossens

(CERN)

Sebastian Rahtz

(ArchaeoInformatica)

1 Slides and L^AT_EX

Many L^AT_EX users want to take advantage of T_EX's high-quality typesetting when they produce overhead slides for a presentation. This facility was originally provided by a separate package, SLI_TE_X, but that had a number of disadvantages:

- it was limited to a set of specially-scaled Computer Modern fonts and it was not easy to adapt to other fonts;
- the user was required to have two separate files, one for control information and the other for the actual slides;
- the control of colour and overlays was limited and crude;
- There was only one 'style' for slides, and writing a different layout (to, say, put a logo on each slide) was not documented.

L^AT_EX users now have a variety of fonts, and vast numbers of styles, to choose from, but SLI_TE_X has lagged behind. When L^AT_EX_{2 ϵ} was released at the end of 1993, this included a simple L^AT_EX package (already available in the New Font Selection Scheme, version 2) to emulate SLI_TE_X without the overhead of a separate macro package. However, there is a much better L^AT_EX package which has been available for some time now—**seminar.sty**; if used in conjunction with a POSTSCRIPT printer, and a set of useful macros called PStricks,² this offers almost every imaginable facility, including:

- ☞ Fancy frames, headers and footers;
- ☞ Landscape and portrait slides in the same document;
- ☞ Coloured text and tables;
- ☞ Interleaving of annotations and slides;
- ☞ Slide 'chapters' and list of slides;
- ☞ Overlays.

seminar is a normal L^AT_EX package which can be used with almost all other L^AT_EX packages (such as those to change font, include graphics etc). Its main job is to produce transparencies, but it can also make accompanying

notes from the same file. It is compatible with AMS-L^AT_EX and L^AT_EX_{2 ϵ} .

2 Using the seminar style

Usage is simple; begin your document in the normal way³ with

```
\documentclass{seminar}
```

or

```
\documentstyle[... options ...]{seminar}
```

The slide environments are

```
\begin{slide}
```

```
...
```

```
\end{slide}
```

```
\begin{slide*}
```

```
...
```

```
\end{slide*}
```

Where `slide` is for landscape slides and `slide*` is for portrait slides. By default, the document is typeset in *landscape* mode, but if you include the `portrait` style option, the document is typeset in portrait mode. *Typesetting* the document in landscape mode is different from *printing* it in landscape mode; you have to worry about the orientation of the page when printing, but with `dvips` this is simple, and taken care of in the local control file described below.

So the default output⁴ from this input:

```
\documentclass{seminar}
\usepackage{times}
\begin{document}
\begin{slide}
My talk is about:
\begin{description}
\item[Cats] Nice furry creatures
which should have a place in
every good home;
\item[Dogs] Nasty barking things
which bite you;
```

¹ Reprint from the Annals of the UK T_EX Users' Group: **Baskerville**, Volume 4.1, Januari 1994.

² The `seminar` package and PStricks are the work of Timothy van Zandt (tvz@princeton.edu).

³ We are assuming L^AT_EX_{2 ϵ} here, just to remind you to upgrade.

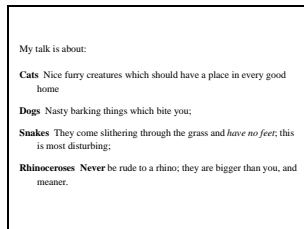
⁴ We have added a style option `output 'times'` so that the output will reduce properly to thumbnails for this article.

```

\item[Snakes] They come slithering
  through the grass and \emph{have
    no feet}; this is most
  disturbing;
\item[Rhinoceroses]
  \textbf{Never} be rude to a
  rhino; they are bigger than you,
  and meaner.
\end{description}
\end{slide}
\end{document}

```

will look like:



Most slides will be no more complicated than this, using standard \LaTeX environments like `itemize`, `enumerate` and `tabular`.

3 Frame styles

A variety of slide framing styles are available, set with the `\framestyle` command; the following are some of the predefined ones (some assume you have a POSTSCRIPT printer), using the `\slideframe` command:

none

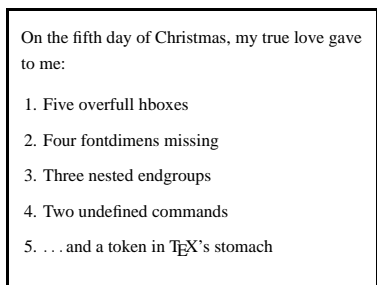
```

On the fifth day of Christmas, my true love gave
to me:

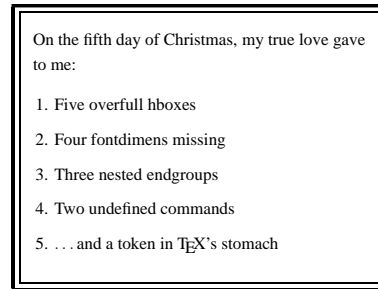
1. Five overfull hboxes
2. Four fontdimens missing
3. Three nested endgroups
4. Two undefined commands
5. ... and a token in  $\TeX$ 's stomach

```

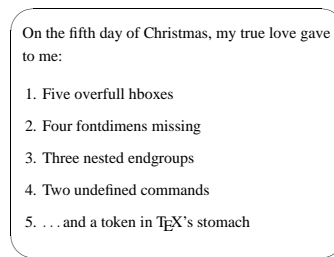
shadow



double

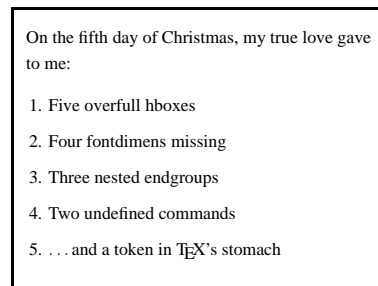


oval

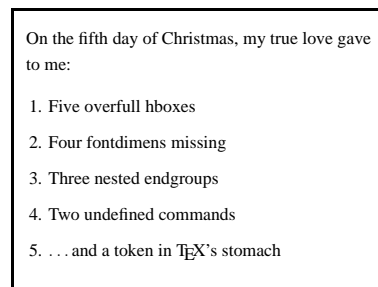


Similarly, a variety of page styles (the headers and footers) are available with the `\pagestyle` command, such as:

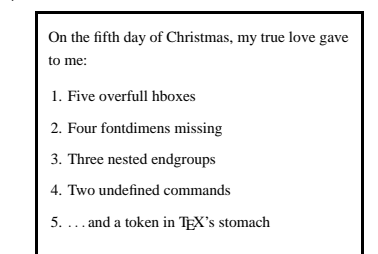
empty



plain



align



Both slide frames and page styles can be customized; for instance, the examples in this paper (eg Figure 1) are suitable for use at the European Particle Physics Laboratory (CERN).

4 Interleaving notes, and selecting subsets

It is easy to intersperse your slides with notes to yourself; these can be simply placed between the `slide` environments or enclosed in a specific `note` environment. You can use any \LaTeX commands in these notes, and include your whole article here if desired. When you want to print the slides, a variety of style options can be used:

- slidesonly** Only the slides are printed;
- notesonly** Only the notes are printed;
- notes** The slides are interleaved with the notes;
- article** The document notes are typeset like a normal \LaTeX paper, and the slides are placed as figures (reduced to half size).

The `\slideplacement` command can be used to affect how slides are placed in the `\article` format; the possible parameters are:

- float** (default) Slides are floated
- float*** Slides are floated, but if two column format is chosen they will span both columns
- here** Slides are placed where they occur in the notes

Further detailed control of the interaction between slides and notes is given in the *User's Manual*.

Selected slides can be included or excluded with the `\onlyslides` or `\noteslides` commands which a parameter of a comma-separated list of slides; this can be numbers, ranges (e.g., 5–10) or \LaTeX `\ref` commands referring to `\label` commands in the slides.

5 Control over slide size, fonts and magnification

There are a great number of parameters by which the user can change any of the following either on a slide-by-slide basis, or for the whole document:

- Slide height and width;
- Top, bottom, left and right margins;
- Text justification (it is ragged right by default);
- Page breaking by varying tolerance of over-running material;
- Inter-line spacing;
- Point size, and choice of fonts.

How to change the default settings is explained in detail in the *User's Guide*.

Because `seminar` works by magnifying pages, sophisticated users should read the manual to see how to deal with setting and changing \TeX dimensions. Most users need not worry about this—in commands like `\epsfig` you should always express your ‘width’ and ‘height’ requests in fractions of the line size anyway.

6 Fonts

As the size of the text on your slides is larger than that for normal text (articles, manuals, etc.), often you will find that

some fonts (especially for mathematics) are not available at the size needed. If you use *dvips*, it tries to generate these fonts but does not always find the necessary Metafont input files to successfully complete the generation of the new font bitmaps. It makes sense, therefore, to use scaleable POSTSCRIPT fonts; the complete Computer Modern family is available in POSTSCRIPT Type1 format, either in the commercial set from Blue Sky or Y&Y, or in (slightly less polished) public domain Paradissa set by Basil Malyshev.

7 The local control file

The `seminar` package always starts by trying to find a file called `seminar.con` on the `TEXINPUTS` path; this gives the user or site an opportunity to conveniently overwrite the defaults. The `seminar.con` file can contain any \LaTeX commands, including inputting style files. Figure 1 was typeset using a `seminar.con` set up for CERN; the contents of this are given below, with explanation of what is being done:

First, we set up landscape macros for the *dvips* driver.

```
\newcommand{\printlandscape}{%
  \special{papersize=297mm,210mm}}
```

We will assume POSTSCRIPT printers, and gain nice POSTSCRIPT effects like rounded box corners; these will need some extra style files:

```
\input semcolor.sty
\input fancybox.sty
```

Since this setup is designed to be used in Europe, where A4 paper is standard, we need to set up some sizes and re-invoke the code to set internal parameters:

```
\def\paperwidth{210mm}
\def\paperheight{297mm}
\input sem-page.sty
\slidewidth 205mm
\slideheight 150mm
\def\slideleftmargin{.8in}
\def\sliderightmargin{.8in}
```

For slide ‘sections’, list of contents, etc we use another style file:

```
\input slidesec.sty
```

This gives us access to various internal commands, some of which are used below. We can also produce a list of slides, in two layouts:

\listofslides	2	\Slidecontents	7
<p>List of Slides</p> <p>Introduction</p> <ul style="list-style-type: none"> 3 Slides and \LaTeX 4 Introducing 'seminar' 5 Normal slide with coloured background and text <p>Frame styles</p> <ul style="list-style-type: none"> 6 Frame styles <p>Text colours and colour tables</p> <ul style="list-style-type: none"> 8 Text colors 9 Colour tests 10 A multi-page coloured table 11 Z schemas built up with overlays 		<p>Introduction</p> <ul style="list-style-type: none"> ✓ • Slides and \LaTeX 3 ✓ • Introducing 'seminar' 4 ✓ • Normal slide with coloured background and text 5 <p>Frame styles</p> <ul style="list-style-type: none"> ✓ • Frame styles 6 <p>Text colours and colour tables</p> <ul style="list-style-type: none"> ⇒ • Text colors 8 • Colour tests 9 • A multi-page coloured table 10 • Z schemas built up with overlays 11 	
January 16, 1994		January 16, 1994	

For slide headings, there is a predefined `\slideheading` command; we will amend this so that it is typeset with a 'shadow'. A `slidechapter` command is also defined:

```
\newcount\@inchap
\@inchap0
\def\l@chapslide#1{%
  \slide@undottedcline{#1}{}%
}
\def\slidechapter#1{%
  \begin{slide}
  \immediate\addtocontents{los}{%
  \protect\l@chapslide{\ignorespaces%
  \protect\bfseries
  \protect\large #1}}%
  \gdef\thechapterheading{#1}%
  \end{slide}
  \@inchap0
}
```

This allows the user to break the slides into groups; the slide chapter title will be given in the bottom right corner with this CERN style.

```
\def\@empty{}
\renewcommand{\makeslideheading}[1]{%
  \gdef\theslideheading{#1}%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty\else
    \begin{Sbox}
      \begin{Bcenter}
        \large\textbf{#1}%
      \end{Bcenter}
    \end{Sbox}
  \centerline{\shadowbox{\TheSbox}}
  \vspace{1ex minus 1ex}
  \fi
}
```

Now the CERN page and frame styles; the plain 'cern' style just places registration '+' marks, and the date:

```
\newpagestyle{cern}%
{ {\color{Black}\small
  \textbf{+} \hfil \today
  \hfil \textbf{+}} }%
{ {\color{Black}\small
  \textbf{+} \hfil \thepage
  \hfil \textbf{+}} }%
```

Whereas the 'cernsections' style has section headings and a logo:

```
\newpagestyle{cernsections}%
{ {\color{Black}\small
  \raisebox{- .5cm}[0cm][0cm]{%
  \epsfig{figure=cernlogo.eps,%
  height=.8cm}}
  \hfil\textbf{\theslideheading}
  \hfil\textbf{\thepage}} }%
{ {\small\color{Black}\today
  \hfil\thechapterheading
  /\inchap
}}%
```

For the slide frames, we define a frame with the word 'CERN' set in a coloured box on the lower left; this is done using the `PStricks` macros (which are automatically included by the 'semcolor' option above). The colour commands are also defined using `PStricks` (the code is not listed here); the names are used in the 'colour' figure.

```
\newslideframe{cern}{ {\SlideFront
  \boxput(-0.7,-1.11){%
  \psframebox[linecolor=black,%
  fillcolor=ForestGreen,%
  fillstyle=solid]{%
  \hbox{%
  {\normalsize
  \sfamily
  \color{Black}CERN}%
  }}{#1}%
  \color{Black}
}}
```

Finally we make sure that each slide starts with the current foreground colour.

```
\def\everyslide{\SlideFront}
\def\theslideheading{}
```

The user sets up slides with the command `\SlideColours`, with two parameters, which are colour names for foreground and background. A synonym is defined for black on white. We have to be a bit careful defining the frame border, because by default it is coloured using the POSTSCRIPT 'setgray' operator, and that might not work with colour separation, so we define an explicit blue frame (for variety).

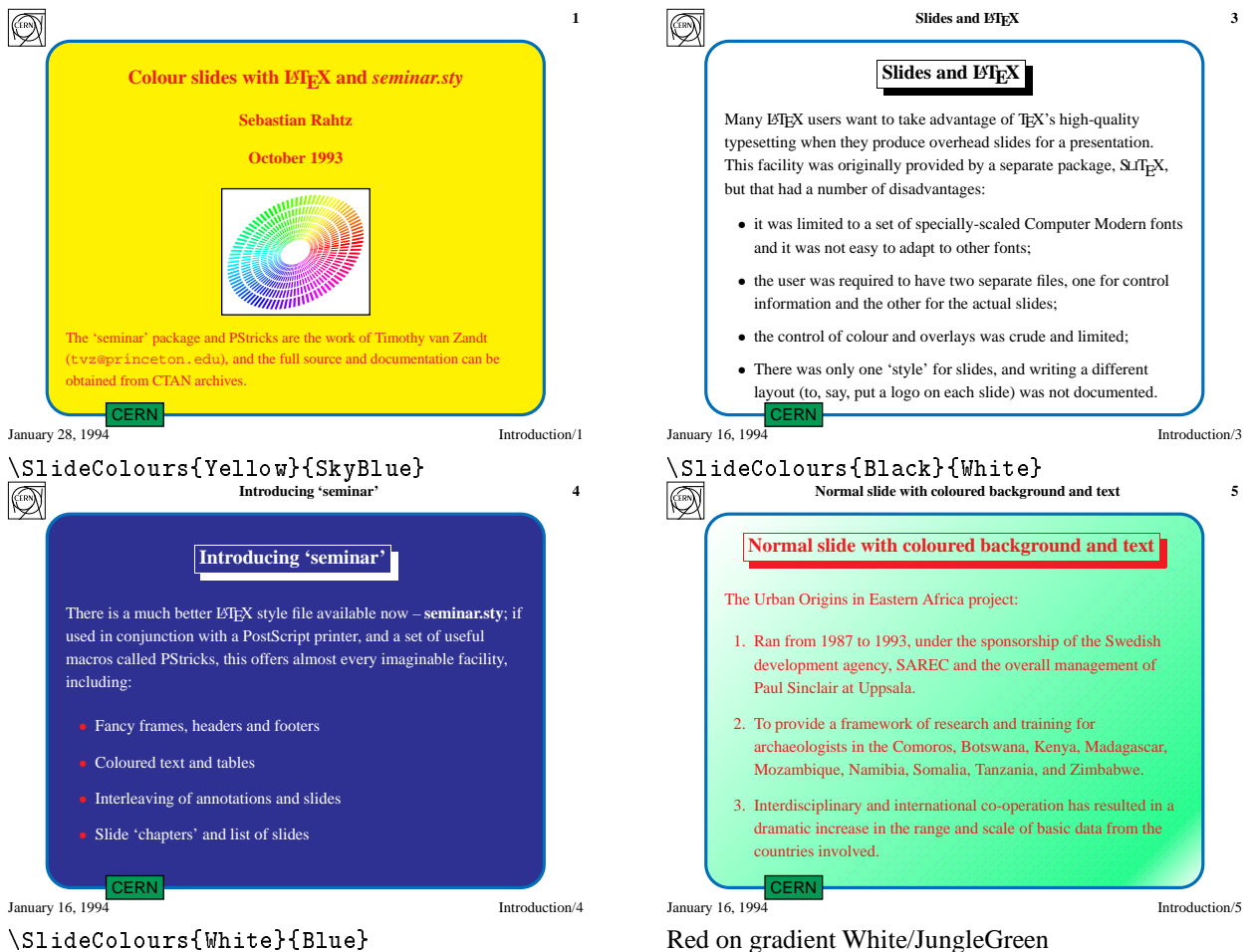


Figure 1: *Colour in slide background and foreground: (simulated in black and white)*

```
\newslideframe{blueframe} [%
\psset{linecolor=NavyBlue,%
linewidth=\slideframewidth,%
framesep=\slideframesep,%
cornersize=absolute,%
linearc=.5cm%
}]{\psframebox{#1}}
\def\SlideColours#1#2{%
\gdef\SlideFront{\color{#1}}%
\slideframe{\Framedefault}%
\slideframe*[\psset{fillcolor=#2,%
fillstyle=solid}]{blueframe}%
}
\def\blackandwhite{%
\SlideColours{Black}{White}%
}
```

The slide defaults will be for a detailed layout and CERN logo, with yellow writing on a blue background:

```
\pagestyle{cernsections}
\slideframe{cern}
\def\Framedefault{cern}
\SlideColours{Yellow}{RoyalBlue}
```

The result of this setup is show in Figure 1; Figure 2 shows how we might combine colour and seminar's overlay com-

mands to highlight the relationship between parts of a program schema.

If you do not like the CERN layout we describe above (or whatever your local system has provided), you can 'roll your own' to various degrees of sophistication. In the simplest case you can just write a file in your local directory:

```
% A do-nothing file seminar.con
\endinput
```

Of course it might be more sensible to build on our work by redefining the commands so that they do what you want. For instance, if you had the seminar.con described above, and wanted to change the default set-up to black on white and use a plain pagestyle and frame, you could add some code in front of the command `\begin{document}`:

```
\documentstyle{seminar}
\pagestyle{plain}
\slideframe{plain}
\renewcommand{\SlideFront}{%
{\color{Black}}
}
\begin{document}
\begin{slide}
...
```

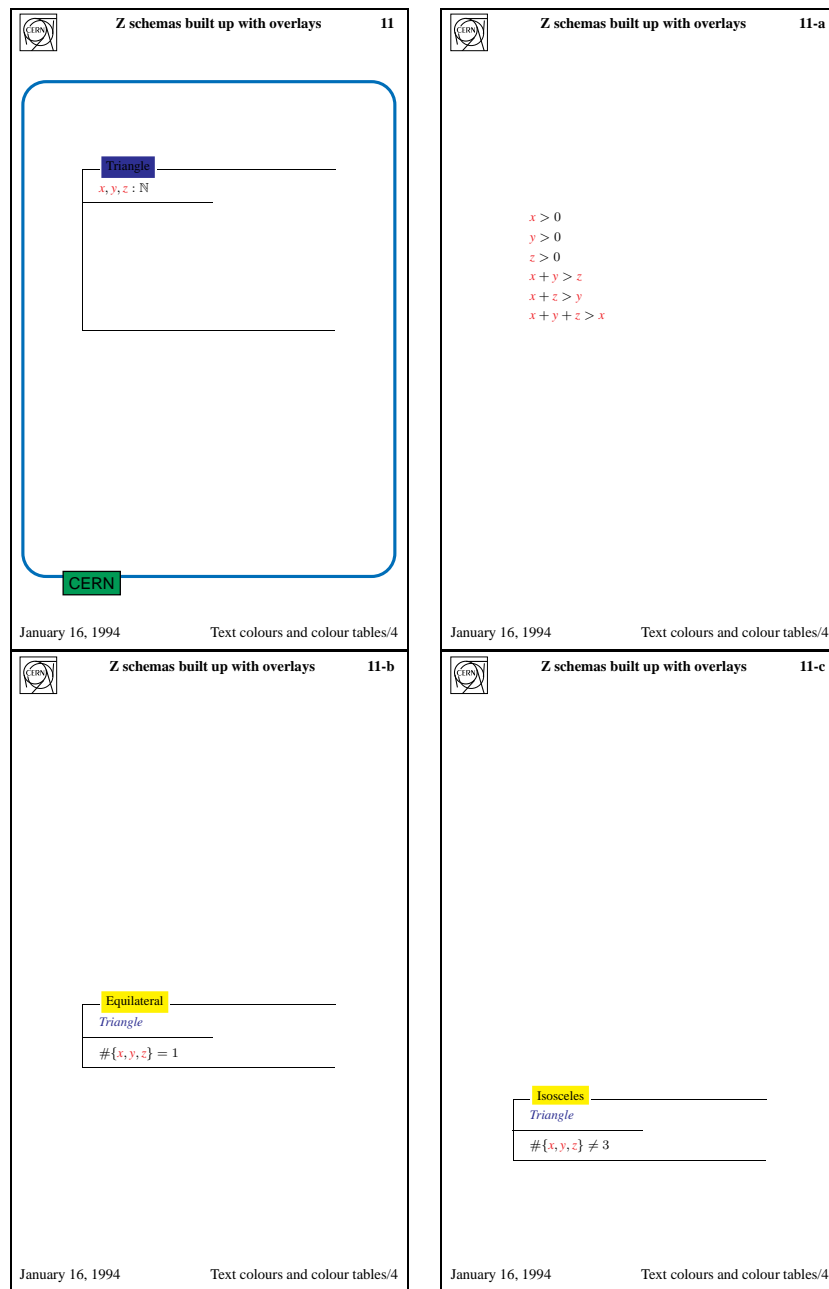


Figure 2: Slide overlays

If you still would like colour backgrounds for your slides by keeping the ‘SlideColours’ interface, but with a plain border and no CERN logos etc., you can use the following set-up:

```
\documentstyle{seminar}
\pagestyle{plain}
\renewcommand{\SlideColours}[2]{%
\renewcommand{\SlideFront}{\color{#1}}%
\slideframe%
[\psset{fillcolor=#2,fillstyle=solid}]%
```

```
{scplain}%
}
\begin{document}
\SlideColours{Red}{Yellow}
\begin{slide}
...
\end{slide}
\end{document}
```

The complete syntax of the many other commands which you have at your disposal is described in the *seminar* user guide; you will probably also want to read the *PStricks* manual!

Making T_EX Work

Norman Walsh

O'Reilly & Associates, Inc.
90 Sherman Street Cambridge, MA 02140
norm@ora.com

20 April 1994

Abstract

This article contains the Table of Contents, Preface, and Chapter 4 of *Making T_EX Work* by Norman Walsh; published by O'Reilly and Associates (ISBN 1-56592-051-1).

No effort has been made to preserve the 'look and feel' of the book in this text, and some simplifications have been performed (particularly with respect to special formatting issues and package names).

Title : Making T_EX Work
Author : Norman Walsh
Publisher : O'Reilly and Associates
Date : April, 1994
ISBN : 1-56592-051-1
Pages : 518
Price : US \$29.95

-- * --

1 Contents 'Making T_EX Work'

1.1 Table of Contents

Preface	xxv
Why Read This Book?	xxv
Scope of This Book	xxvi
Conventions Used in This Book	xxviii
How to Get T _E X	xxix
Where Are the Files?	xxx
Getting Software Without FTP	xxx
Getting Examples From This Book	xxx1
FTP	xxx1
FTPMAIL	xxx2
BITFTP	xxx2
UUCP	xxx3
Gopher	xxx3
Versions of T _E X	xxx3
Implementations and Platforms	xxx4
UNIX	xxx4
MS-DOS	xxx4
OS/2	xxx5
Macintosh	xxx5
We'd Like to Hear From You	xxx5
Acknowledgments	xxx6

I: An Introduction to T_EX	1
1 The Big Picture	3
What Is T _E X?	3
T _E X for Beginners	4
Boxes and Glue	5
Control Sequences	6
Special Characters	8
Text Formatting Versus Word Processing	11
What About Desktop Publishing?	13
What About troff?	14
What About SGML?	15
How T _E X Works	15
T _E Xing a Simple Document	15
Editing your document	16
Running T _E X	16
Using macros	17
Using fonts	18
Previewing or printing T _E X documents	18
T _E Xing More Complex Documents	19
Building a Table of Contents	20
Figure References	20
Indexes and Glossaries	21
Bibliographies	21
Special Things	22
2 Editing	23
What Can an Editor Do?	24
Quoting Automatically	24
Matching Braces	25
Inserting Common Control Sequences	25
Running T _E X Automatically	25
GNU Emacs	26
Starting T _E X Mode	26
Typing in Emacs	27
Running T _E X Automatically	27
AUC-T _E X	27

Starting AUC- \TeX	28	Building the Extended Plain \TeX format	72
Typing in AUC- \TeX	28	$\LaTeX_{2\epsilon}$ Versus \LaTeX	72
AUC- \TeX Outline Mode	31	$\LaTeX_{2\epsilon}$	73
Intelligent Paragraph Reformatting	31	Building the $\LaTeX_{2\epsilon}$ format	74
Multi-file Documents	31	\LaTeX	75
Running \TeX	32	Building the \LaTeX format with the OFSS	76
Finding Errors	32	AMST \TeX	76
Multi-Edit	33	Building the AMST \TeX format	77
Setting Up \TeX Support	33	AMSL \LaTeX	79
Typing in Multi-Edit	34	Building the AMSL \LaTeX format	79
Running \TeX	35	Lollipop	79
Brief	35	Building the Lollipop format	80
MicroEMACS	35	\TeX info	81
epm: OS/2's Enhanced Editor	35	Other Formats	83
Other Tools	36	Special-purpose Formats	84
\TeX as a Compiler	37	Slit \TeX	87
Processing a Document	38	Foil \TeX	87
Spellchecking	41	Seminar	88
ispell	41	\TeX in Color	88
amSpell	41	Setting Up Color	88
Revision Control	42	Using Color	89
\TeX Shells	43	Now I've Got Color, but I Need Black and White!	90
\TeX Shell	43	Color Under $\LaTeX_{2\epsilon}$	90
\TeX Pert	44	Color Is Subtle	90
4 \TeX	44	Further Reading	91
PM \TeX	45		
\TeX it	45		
3 Running \TeX	47	II: Elements of a Complex Document	93
What Do You Run?	47		
What Files Does \TeX Need?	48	5 Fonts	95
Pool Files	48	What \TeX Needs To Know	96
Format Files	49	Selecting a Font in \TeX	97
User Files	49	Which Character Is Which?	98
The Command Line	51	The Issue of Size	98
Command-line Options	51	Expressing Design Size in \TeX	99
Command-line Cautions	52	Expressing Magnification in \TeX	100
Misinterpretation of the command line	52	Standard Magnifications	100
Filenames and \TeX	53	Where Do TFM Files Come From?	101
Multiple documents	54	The New Font Selection Scheme	102
\TeX Without Options	55	Selecting Fonts with the New Font Selection Scheme	104
What About Errors?	56	Low-level interface to NFSS	105
Log Files	58	Defining Fonts with NFSS2	106
Interpreting \TeX Error Messages	59	Declaring a family	107
Naming a File \TeX Cannot Find	59	Declaring shapes	107
Misspelling a Control Sequence Name	60	Storing Font Definitions	109
Failure to Close an Environment	60	Changing the Defaults	109
Math Outside of Math Mode	61	NFSS Pitfalls	110
Missing Fonts	62	PostScript Fonts Under NFSS	111
Everything Else	62	Adjustments to Scale	111
The Question Mark Prompt	62	When Things Go Wrong	112
		When \TeX Complains	112
		When the DVI Driver Complains	113
		Can't find PK file	113
		Accents don't work or the wrong characters are printed	114
		Printer prints the right characters but the wrong font	114
4 Macro Packages	65		
Installation: Making Format Files	67		
Hyphenation Patterns	68		
General-purpose Formats	68		
Plain \TeX	70		
Building the Plain \TeX format	71		
Extended Plain \TeX	71		

Encoding Vectors	115	xfig	162
Virtual Fonts	116	idraw	164
Automatic Font Generation by DVI Drivers	118	tgif	165
Math Fonts in T _E X	120	bitmap/pixmap	165
Concrete Examples	121	Other Bitmap Editors	166
METAFONT Fonts	121	texcad/xtexcad	167
PostScript Type 1 Fonts	121	Screen Dumps	168
Using a new PostScript font in T _E X (for PostScript printers)	123		
Using a new PostScript font in T _E X (for non-PostScript devices)	126		
HP LaserJet Softfonts	127		
TrueType Fonts	128		
6 Pictures and Figures	129	7 International Considerations	171
Different Kinds of Images	129	Typesetting in Any Language	171
Bitmapped Images	130	Reading Input Files	172
Scalable Images	130	Appearances can be deceiving	174
Device Independence Revisited	131	Changing the Rules	174
Using Only T _E X	131	Printing the Result	177
Plain T _E X	132	The Babel Styles	178
L ^A T _E X	132	Building Multilingual Babel Formats	179
The epic and eepic styles	134	T _E X Pitfalls	180
The bezier style	135	Very Complex Languages	181
Other styles	136	Japanese	181
Other approaches	136	ASCII Nihongo T _E X	181
PiCT _E X	137	NTT J _T E _X	182
XY-pic	139	Poor Man's Japanese T _E X	182
DraT _E X	141	JemT _E X	184
Using a Little Help	144	Chinese	185
MFPic	144	Poor Man's Chinese T _E X	185
Fig2MF	144	Arabic	185
METAFONT	145	Hebrew	186
Using a Little More Help	145		
PSTricks	145	8 Printing	189
T _E Xdraw	145	Printing Fonts	189
tpic	145	Built-in Fonts	190
Using a Lot of Help	146	Font substitution in emT _E X	190
Electronic Cut-and-paste	146	Font substitution in <i>dvips</i>	190
Scalable Image Formats	146	Font substitution in other drivers	191
Bitmapped Image Formats	149	External Fonts	191
DVI driver <code>\special</code>	149	Bitmapped Fonts	192
bm2font	149	Scalable Fonts	195
pbmtpk	149	Font Printing Pitfalls	195
Inserting PostScript Images into T _E X	150	Printing Pictures and Figures	196
Manipulating Images	151	Unsolvable Problems	196
Image Magick	153	Solvable Problems	196
PBMplus	154	Pictures Using Only T _E X	197
xv	155	METAFONT Figures	197
xloadimage	156	Scalable Images	197
Image Alchemy	156	Bitmap Images	197
ColorView	157	Selected Drivers	197
Jpeg4	157	emT _E X Drivers	198
pmjpeg	157	dviIj2	200
txt2pcx	157	dvips	200
Ghostscript	157	DVILASER/HP	201
GoScript	161	DVILASER/PS	203
hp2xx	161	PTI Laser/HP and PTI Jet	205
Image Editors	162	PTI Laser/PS	205
		dvipsone	206
		9 Previewing	209
		Previewing Under X11	210
		Previewing with xdvi	210
		Previewing with X _T E _X	213

Previewing with Ghostscript	214	bibclean	260
Previewing with em \TeX	215	citetags	261
Previewing with dvivga	217	citefind	261
\TeX Preview	219	bibextract	261
dvideo	221	lookbibtex	261
PTI View	223	bibdestringify	261
Previewing Under Windows	224	edb	261
dvimswin	224	Emacs Bib \TeX mode	261
dviwin	225	bibview	262
wdviwin	226	xbibtex	262
DVIWindo	226	bibdb	263
Previewing on a TTY	229	Tib	264
dvi2tty	229	Making Indexes	265
dvgt/dvitovdu	230	Index Entries	265
crudetype	230	Index Format	266
10 Online Documentation	233	Special-purpose styles	266
Something Is Lost	234	Making Glossaries	266
\TeX info	235		
\LaTeX 2HTML	236	III: A Tools Overview	269
Lame \TeX	238		
latex2hy	239	13 Non-commercial Environments	271
detex	240	Web2C	271
dvispell	240	em \TeX	272
11 Introducing METAFONT	241	Where to Start	274
What to Run?	242	Unpacking the Archives	274
What Files Does METAFONT Need?	243	Setting Up the Environment	275
Pool Files	243	Testing the Installation	275
Base Files	243	Installing Fonts	276
METAFONT Programs	244	Testing font installation	276
Command-line Options	244	Making em \TeX Smaller	277
Building a Base File	245	Deleting extra documentation	277
Running METAFONT	245	Deleting extra executables	277
Picking a Mode	245	Deleting extra printer drivers	279
Proofing mode	246	Basic Customization	280
Selecting a Size	246	Environment variables	280
Making a GF Font	247	Setting up configuration files	282
Making a PK Font	247	Installing the Beta Test Versions of em \TeX	282
What About Errors?	248	Running em \TeX 386 in Windows	283
Other Errors	251	Automatic Font Generation with em \TeX	284
Output at Very High Resolutions	251	Installing em \TeX on a Network	284
Output at Very Low Resolutions	252	texas	285
12 Bibliographies, Indexes, and Glossaries	253	sb \TeX	286
Bib \TeX	253	g \TeX	287
How Bib \TeX Works	254	14 Commercial Environments	289
Building a Bibliography Database	255	$\mu\TeX$ by ArborText	289
Database entries	255	Y&Y \TeX	290
Entry types	256	Textures	291
Abbreviations	257	Turbo \TeX	291
Preamble	257	PCT \TeX	292
Comments	258	PCT \TeX For Windows	294
Special characters	258	Scientific Word	295
Bibliography Styles	258	15 \TeX on the Macintosh	299
Special-purpose styles	259	CMac \TeX	299
Bibliography Database Tools	259	Direct \TeX	300
bibsrt	260	Oz \TeX	301
biblook	260	Textures	302

Other Tools	304	2-11 PMT _E X	46
Alpha	304	2-12 T _E Xit	46
B _B Edit	304	4-1 Sample page	69
BSR Font Tools	307	4-2 AMS sample page	78
Excalibur	307	4-3 T _E Xinfo sample page	82
HyperBibT _E X	307	4-4 Online documentation produced by MakeInfo	83
MacGS	307	4-5 Caffeine by ChemT _E X	85
dvidvi	307	4-6 A lithium cation rendered by <i>ChemStruct</i>	86
MacDVicopy	307	4-7 A little Mozart.	86
MacBibT _E X	310	5-1 “fi” as two characters and as a ligature	97
MacMakeIndex	310	5-2 “We” unkernd and kernernd	97
16 T_EX Utilities	311	5-3 The Computer Modern Roman letter “R” at 150pt: (a) from a 5pt design; (b) from a 17pt design	99
List of Tools	312	5-4 How T _E X uses a virtual font	117
A Filename Extension Summary	335	6-1 An example diagram in Plain T _E X	132
B Font Samples	345	6-2 A parallelogram in L ^A T _E X	133
Font Encodings	345	6-3 A figure created with (a) epic, and (b) eepic	135
Font Samples	352	6-4 Several bezier curves created with the L ^A T _E X bezier style	136
C Resources	391	6-5 Sample diagrams using Pict _E X	137
TUG: The T _E X Users Group	391	6-6 An X _Y -pic diagram	140
Other User Groups	392	6-7 An DraT _E X diagram	142
T _E X Software	392	6-8 A PostScript bounding box example	148
T _E X Shells	394	6-9 An example of an encapsulated figure	151
Editors	394	6-10 Another example of an encapsulated figure (resized and rotated)	152
Macro Packages (Formats)	396	6-11 Editing an image with xfig	163
Styles and Macros	398	6-12 Editing an image with idraw	164
Styles That Produced This Book	400	6-13 Editing an image with tgif	166
Utilities	402	6-14 An example of bitmap editing an icon	167
Miscellaneous	419	6-15 A text mode screen dump.	169
D Long Examples	421	7-1 Character mapping example	175
Bibliography	465	7-2 Poor Man’s Japanese	183
Index	469	7-3 <i>JemT_EX</i> sample	184
		7-4 Poor Man’s Chinese with traditional char- acters	185
		7-5 Poor Man’s Chinese with simplified char- acters	186
		7-6 Arabic text typeset with ArabT _E X	187
		7-7 Hebrew	188
		8-1 Resolution of the bitmap and the device changed simultaneously	193
		8-2 Resolution of the bitmap changed while device held constant	193
		8-3 Previewing and printing with emT _E X	199
		9-1 How Previewing with x _{dvi} Works	211
		9-2 Previewing with x _{dvi}	212
		9-3 How Previewing with XT _E X Works	213
		9-4 Previewing with XT _E X	214
		9-5 Previewing with Ghostview	216
		9-6 Previewing with emT _E X’s dviscr	217
		9-7 Previewing with emT _E X’s dvipm	218
		9-8 Previewing with dvivga	219
		9-9 Previewing with ArborText’s Previewer	220
		9-10 Previewing with <i>TurboT_EX dvideo</i> (using limited selection of fonts)	222
		9-11 Previewing with Personal T _E X’s Previewer	223

— * —

1.2 List of Figures

1-1 The Letters “g” and “h” inside their boxes.	5
1-2 Example figure produced by both T _E X and <i>troff</i>	15
1-3 A high-level view of T _E X	16
1-4 High-level view of T _E X including more detail	17
1-5 T _E Xing a More Complex Document	20
2-1 Extension setup in Multi-Edit	33
2-2 L ^A T _E X as a compiler in Multi-Edit	34
2-3 Editing a file with <i>MicroEMACS</i>	36
2-4 Microsoft Windows online help for L ^A T _E X	37
2-5 Editing a T _E X document with epm under OS/2	38
2-6 Edit settings control word delimiters	41
2-7 Spellchecking a document with amSpell	42
2-8 T _E XShell	44
2-9 T _E XPert	45
2-10 4T _E X	46

9-12	Previewing with <code>dvimswin</code>	224	8-4	Other <i>DVILASER/PS</i> Utilities	204
9-13	Previewing with <code>dviwin</code>	225	8-5	Other <code>dvipsone</code> Utilities	207
9-14	Previewing with Turbo \TeX 's <code>wdviwin</code>	226	9-1	Common Previewers	210
9-15	Previewing with Y&Y's <code>DVIWindo</code>	227	9-2	\TeX Preview Utilities	221
9-16	<code>DVIWindo</code> preview much enlarged	228	9-3	<code>DVIWindo</code> Utilities	229
9-17	Previewing with <code>dvgt</code> under Tektronics 4010 emulation	231	11-1	Some Popular METAFONT Fonts on the CTAN Archives	242
10-1	\TeX info sample page	237	12-1	Types of Entries with Required and Op- tional Fields	256
10-2	Online documentation produced by <code>MakeInfo</code>	238	13-1	Summary of the <code>em\TeX</code> Distribution	273
11-1	A figure eight created with METAFONT	248	13-2	Fonts Libraries Available for <code>em\TeX</code> on CTAN	276
12-1	Editing an article entry with <code>bibview</code>	262	13-3	Other <code>em\TeX</code> Executables	278
12-2	Editing an article entry with <code>xbibtex</code>	263	13-4	The Beta Test Files for <code>em\TeX</code>	282
12-3	Editing an article entry with <code>bibdb</code>	264	14-1	Turbo \TeX Programs	291
14-1	The <code>PCT\TeX</code> menu system	293	14-2	Turbo \TeX METAFONT Programs	292
14-2	<code>PCT\TeX</code> For Windows editing and preview- ing a \LaTeX document	294	15-1	Summary of the <code>CMac\TeX</code> Distribution at CTAN	300
14-3	<code>PCT\TeX</code> For Windows TrueType font metric builder	295	15-2	Summary of the <code>Oz\TeX</code> Distribution at CTAN	301
14-4	Editing a document with <i>Scientific Word</i>	296	15-3	Font Tools in the BSR package	307
14-5	Preview of the <i>Scientific Word</i> document shown in Figure 14-4	297	16-1	The Public Doman DVI Driver Family	315
15-1	Editing, previewing, and typesetting in Textures	303	16-2	RTF Translators	324
15-2	<i>Alpha</i> editing the fonts chapter from this book	305	16-3	Additional Scripts in PS Utils	330
15-3	<i>BEdit</i> editing the fonts chapter from this book	306	B-1	The Computer Modern Roman Font En- coding	346
15-4	<i>Excalibur</i> spellchecking the fonts chapter from this book	308	B-2	The Computer Modern Math Italic Font Encoding	347
15-5	The <i>HyperBIB\TeX</i> view of a bibliographic database	309	B-3	The Cork Font Encoding	348
			B-4	The Adobe Standard Font Encoding	350

-- * --

-- * --

1.3 List of Tables

1-1	Special Characters in Plain \TeX	9
1-2	How to Typeset Special Characters	9
2-2	AUC- \TeX Math Operators in Mathematics Minor-mode	30
2-3	\TeX Modes of Interaction	40
5-1	Standard Weight and Width Designations	103
5-2	Weight and Width Are Combined to Form Series	103
5-3	Standard Abbreviations of Font Shape	104
5-4	User-level Font Selection Control Se- quences in NFSS2	104
5-5	Encoding Schemes Supported by NFSS2	106
5-6	Default Fonts	109
6-1	Graphics Manipulation Packages	152
6-6	Graphics Editing Packages	162
7-1	Standard Control Sequences for Symbols from Other Character Sets	173
7-2	New Control Sequences Proposed by TWGMLC	173
7-3	Language Switch Macros Proposed by TUG	176
8-1	Common DVI Drivers	198
8-3	Other <i>DVILASER/HP</i> Utilities	202

1.4 List of Examples

1-1	An Example of a \TeX Document	6
1-2	A typical bibliography database entry	21
2-1	A Simple Driver File	31
2-2	Local Variables in an Emacs Buffer	32
2-3	A Contrived Error	38
2-4	Revision Control Macros for \TeX Docu- ments Using RCS	42
3-1	The Document <code>BADFON.TEX</code>	57
4-1	Plain \TeX Input	70
4-2	$\LaTeX_{2\epsilon}$ Input	74
4-3	\LaTeX Input File	76
4-4	<code>AMST\TeX</code> Input File	77
4-5	Lollipop Input File	79
4-6	\TeX info Input	81
4-7	The <code>Chem\TeX</code> Source for Caffeine	85
4-8	The <code>ChemStruct</code> Source for the Lithium Cation	85
4-9	The <code>Music\TeX</code> Source for Figure 4-7	86
5-1	Font-shape Declaration with NFSS2	108
5-2	Font-shape declaration with NFSS2 (sim- plified)	108
5-3	The <code>PSto\TeXfont</code> script	119
6-1	The Input for the Plain \TeX Diagram	132
6-2	The \LaTeX Input for Figure 6-2	133
6-3	The <code>epic</code> Input for Figure 6-3	134

6-4	The Input for Figure-4	135
6-5	PiCT _E X Input for Figure 6-5	137
6-6	The XY-pic Input for Figure 6-6	140
6-7	The DraT _E X Input for Figure 6-7	142
6-8	Converting Encapsulated PostScript to a Bitmap with <i>Ghostview</i>	159
6-9	Script for Inserting a Captured Text Screen	169
7-1	A Sample Multilingual Document Using English and French	179
10-1	T _E Xinfo Commands	235
11-1	The Code for the Figure Eight	248
12-1	A Sample BibT _E X entry	255
12-2	A Tib style database entry	264
D-1	MakeTeXPK.pl	421
D-2	MakeTeXTFM.pl	433
D-3	dvidxx.btm	442
D-4	makepk.btm	447
D-5	enc-afm.pl	450
D-6	TeXtoXfont	454
D-7	PStoXfont	458
D-8	txt2verb.pl	460

— * —

2 Preface ‘Making T_EX Work’

T_EX is a tool for creating professional quality, typeset pages of any kind. It is particularly good, perhaps unsurpassed, at typesetting mathematics; as a result, it is widely used in scientific writing. Some of its other features, like its ability to handle multiple languages in the same document and the fact that the content of a document (chapters, sections, equations, tables, figures, etc.) can be separated from its form (typeface, size, spacing, indentation, etc.) are making T_EX more common outside of scientific and academic circles.

Designed by Donald Knuth in the late 1970s, more than a decade of refinement has gone into the program called ‘T_EX’ today. The resulting system produces publication-quality output while maintaining portability across an extremely wide range of platforms.

Remarkably, T_EX is free. This fact, probably as much as any other, has contributed to the development of a complete ‘T_EX system’ by literally thousands of volunteers. T_EX, the program, forms the core of this environment and is now supported by hundreds of tools.

2.1 Why Read This Book?

This book is for anyone who uses T_EX. Novices will need at least one other reference, because this book does not describe the nuts and bolts of writing documents with T_EX in any great detail.

If you are new to T_EX, there is much to learn. There are many books that describe how to use T_EX. However, the focus of this book is mostly at a higher level. After digesting Chapter 1, *The Big Picture*, you should be able to proceed through the rest of the book without much diffi-

culty even if you have never seen T_EX before. So, if you are a system administrator interested in learning enough about these programs to install and test them for your users, you should be all set. If you are interested in learning how to write documents with T_EX, this book will be helpful, but it will not be wholly sufficient.

Why do you need this book at all? Although many individual components of the T_EX system are well documented, there has never before been a complete reference to the whole system. This book surveys the entire world of T_EX software and helps you see how the various pieces are related.

A functioning T_EX system is really a large collection of programs that interact in subtle ways to produce a document that, when formatted by T_EX, prints the output you want. All the different interactions that take place ultimately result in less work for you, the writer, even though it may seem like more work at first. Heck, it may *be* more work at first, but in the long run, the savings are tremendous.

Many books about T_EX refer the reader to a ‘local guide’ for more information about previewing and printing documents and what facilities exist for incorporating special material into documents (like special fonts and pictures and figures). In reality, very few local guides exist.

The T_EX environment is now mature and stable enough to support a more ‘global guide.’ That is what this book attempts to be. It goes into detail about previewing and printing, about incorporating other fonts, about adding pictures and figures to your documents, and about many other things overlooked by other books.

Because fonts play a ubiquitous role in typesetting, this book is also about METAFONT, the tool that Donald Knuth designed for creating fonts.

2.2 Scope of This Book

Here’s how the book is laid out.

Part I: An Introduction to T_EX.

Chapter 1, *The Big Picture*. If you don’t know anything about T_EX at all, this chapter will help you get started. If you’re a system administrator charged with the task of installing and maintaining T_EX tools, you’ll get enough information to do the job.

Chapter 2, *Editing*. An overview of some environments that can make working with T_EX documents easier. It describes some editors that ‘understand’ T_EX, how to integrate T_EX into a ‘programmer’s editor,’ spellchecking, revision control, and other aspects of T_EXnical editing.

Chapter 3, *Running T_EX*. The mechanics of running T_EX, the program. It discusses what things T_EX needs to be able to run, how to start T_EX, command-line options, leaving T_EX, and recovery from errors.

Chapter 4, *Macro Packages*. Overview of T_EX macro packages. This chapter describes how to make a format file, the major general-purpose writing packages, some special-

purpose writing packages, how to make slides for presentations, and how to handle color in T_EX.

Part II: Elements of a Complex Document.

Chapter 5, *Fonts*. This chapter explores the issues that need to be addressed when using fonts. Many of these issues are not particularly T_EX-related, but T_EX is very flexible, and it's important to understand the tradeoffs that must be made. This chapter also examines some T_EX-specific issues: font selection, files that T_EX needs, automatic font generation, and virtual fonts.

Chapter 6, *Pictures and Figures*. How many ways are there to include pictures and figures in a T_EX document? Lots and lots of ways. This chapter examines them all.

Chapter 7, *International Considerations*. T_EX is well qualified to do international typesetting. This chapter looks at the issues that are involved: representing international symbols in your input file, what T_EX produces, getting the right fonts, multiple languages in the same document, and macro packages and style files that solve some of these problems. Some strategies for dealing with very difficult languages (like Japanese and Arabic) are also explored.

Chapter 8, *Printing*. What goes in has to come out. This chapter tells the what, where, why, and how of printing your documents.

Chapter 9, *Previewing*. Save paper; preview your documents before you print them.

Chapter 10, *Online Documentation*. Online documentation is becoming increasingly popular. This chapter explores different ways that both typeset and online documentation can be produced from the same set of input files.

Chapter 11, *Introducing METAFONT*. Sometimes it is necessary or desirable to create a special version of a standard T_EX font. Maybe you really need the standard 10pt font at 11.3pt. This chapter will tell you how to work with existing METAFONT fonts. It *won't* tell you how to create your own fonts; that's a whole different story.

Chapter 12, *Bibliographies, Indexes, and Glossaries*. Maintaining a bibliographic database can be a great timesaver. This chapter looks at the BibT_EX program and other tools for building and using bibliographic databases. It also discusses the creation of indexes and glossaries.

Part III: A Tools Overview.

Chapter 13, *Non-commercial Environments*. Many T_EX environments are freely available. This chapter describes public domain, free, and shareware versions of T_EX.

Chapter 14, *Commercial Environments*. A large, complex system like T_EX can be overwhelming (although I hope less so after you read this book ;-). One of the advantages of selecting a commercial implementation of T_EX is that some form of customer support is usually provided. Still other commercial implementations offer features not

found in any free releases. This chapter describes several commercial T_EX releases.

Chapter 15, *T_EX on the Macintosh*. Many issues discussed in this book apply equally to all platforms, including the Macintosh platform, but the Mac has its own special set of features. This chapter looks at some versions of T_EX and other tools designed specifically for use on the Mac.

Chapter 16, *T_EX Utilities*. This chapter lists many of the the utilities available in the CTAN archives and provides a brief description of what they do.

Appendix A, *Filename Extension Summary*. Lots of files can be identified by their extensions. This appendix lists the extensions that are most often seen in conjunction with T_EX and describes what the associated files contain.

Appendix B, *Font Samples*. Examples of many METAFONT fonts available from the CTAN archives.

Appendix C, *Resources*. A complete list of the resources described in this book.

Appendix D, *Long Examples*. This appendix contains examples (scripts, batch files, programs) that seemed too long to place in the running text.

Bibliography. Where I learned what I know. Also, where you can look for more information about specific topics.

2.3 Conventions Used in This Book

The following typographic conventions are used in this book:

Italic is used for filenames, directories, user commands, program names, and macro packages (except where the distinctive logo type is used). Sometimes italics are used in the traditional way for emphasis of new ideas and important concepts.

`Typewriter` is used for program examples, FTP sites, T_EX control sequences, and little bits of T_EX syntax that appear in running text (for example, typewriter text in a reference to the `\LaTeX picture` environment is a clue that `\LaTeX` literally uses the word 'picture' to identify this environment).

`Typewriter Bold` is used in examples to show the user's actual input at the terminal.

`Typewriter Italic` identifies text that requires a context-specific substitution. For example, `filename` in an example would be replaced by some particular filename.

Footnotes are used for parenthetical remarks. Sometimes, lies are spoken to simplify the discussion, and the footnotes restore the lie to the truth.¹

Filename extensions, like `.tex` in `book.tex`, are shown in uppercase letters when referring to a particular type of file. For example, a T_EX Font Metric or TFM file would be `somefile.tfm`. The actual extension of the file may be different (upper or lowercase, longer or shorter) depending on your operating environment.

¹ And sometimes they don't ;-)

When the shell prompt is shown in an example, it is shown as a dollar sign, \$. You should imagine that this is your system's prompt, which might otherwise be %, C>, [C:\], or a dialog box.

When spaces are important in an example, the ‘ ’ character is used to emphasize the spaces. Effective, isn't it?

In some places, I refer to specific keys that you should press. When it's important that I mean pressing particular keys and not typing something, I emphasize the keys. For instance, an example that includes Enter means that you should literally press the Enter or Return key. The sequence Ctrl-D means that you should press and hold the ‘Control’ and ‘d’ keys simultaneously. Control-key combinations aren't case sensitive, so you needn't press the shift key.

2.4 How to Get T_EX

T_EX and the other programs mentioned in this book are available from a number of places. It's impossible to list all of the places where you might find any given tool, but there is one place where you will almost certainly find every tool: the Comprehensive T_EX Archive Network (CTAN).

This network is a fully-mirrored anonymous FTP hierarchy on three continents. Always use the FTP site that is geographically closest to you. The following table lists the current members of CTAN as of July, 1993:

Geographic Location	Site	IP Address	Top Level Directory
United States	ftp.shsu.edu	192.92.115.10	/tex-archive
England	ftp.tex.ac.uk	131.151.79.32	/tex-archive
Germany	ftp.uni-stuttgart.de	129.69.8.13	/tex-archive

You may also access the CTAN archives by electronic mail if you do not have FTP access. For up-to-date instructions about the mail server, send the single-line message help to fileserv@shsu.edu.

2.4.1 Where Are the Files?

Every CTAN mirror site has the same well-organized directory structure. The top-level directory also contains a complete catalog of current files organized by name, date, and size. The catalogs are named *FILES.byname*, *FILES.bydate*, and *FILES.bysize*, respectively, in the top level directory. The top-level directory contains the following subdirectories:

The archives at [ftp.shsu.edu](ftp://ftp.shsu.edu) and [ftp.tex.ac.uk](ftp://ftp.tex.ac.uk) also support *gopher* access to the archives. The UK *gopher* supports indexed access to the archives. A World Wide Web (hypertext) interface to the archives is available from:

<http://jasper.ora.com/CTAN/ctan.html>

This interface includes brief descriptions of many packages and the ability to perform keyword and date searches.

Directory	Description of Contents
<i>tools</i>	Archiving tools (<i>unzip</i> , <i>tar</i> , <i>compress</i> , etc.)
<i>biblio</i>	Tools for maintaining bibliographic databases
<i>digests</i>	Electronic digests (T _E Xhax, UKT _E X, etc.)
<i>info</i>	Free documentation, many good guides
<i>dviware</i>	Printing and previewing software
<i>fonts</i>	Fonts for T _E X
<i>graphics</i>	Software for working with pictures and figures
<i>help</i>	Online help files, etc.
<i>indexing</i>	Indexing and glossary building tools
<i>language</i>	Multi-national language support
<i>macros</i>	Macro packages and style files
<i>misc</i>	Stuff that doesn't fit in any other category
<i>support</i>	Tools for running and supporting T _E X
<i>systems</i>	OS-specific programs and files
<i>web</i>	Sources for T _E X programs (in Web)

2.4.2 Getting Software Without FTP

The electronic alternatives to FTP, described in the section ‘Getting Examples From This Book’ of this chapter are also viable alternatives for getting software from the CTAN archives.

In addition, there are a number of ways to get distributions through nonelectronic channels. The names and addresses of these sources are listed in Appendix C, *Resources*.

You can get many of the popular T_EX distributions on diskette from the T_EX Users Group (TUG). Emacs, *Ghostscript*, and other packages by the Free Software Foundation (FSF) are available on tape directly from the FSF. You may also find large bulletin board systems that support T_EX (for example, Channel1 in Cambridge, MA)

2.4.3 Getting Examples From This Book

All of the substantial code fragments and programs printed in this book are available online. The examples in this book are all in *Perl*, a language for easily manipulating text, files, and processes. I decided to use *Perl* simply because it is available for every platform discussed in this book. It is the only ‘universal’ scripting language that will work under MS-DOS, OS/2, UNIX, and the Macintosh. All of the scripts in this book can be converted to a different scripting language (the various UNIX shells or something like 4DOS's extended batch language for MS-DOS and OS/2) if you prefer. I've tried to write the *Perl* scripts in a straightforward way so that any given task won't be too difficult.

The examples are available electronically in a number of ways: by FTP, FTPMAIL, BITFTP, and UUCP. The cheapest, fastest, and easiest ways are listed first. If you read from the top down, the first one that works is probably the best. Use FTP if you are directly on the Internet. Use FTPMAIL if you are not on the Internet but can send and receive electronic mail to Internet site (this includes CompuServe users). Use BITFTP if you send electronic mail via BITNET. Use UUCP if none of the above work.

NOTE: The examples were prepared using a UNIX system. If you are running UNIX, you can use them without modification. If you are running on another platform, you may

need to modify these examples to correct the end-of-line markers. For example, whereas under UNIX every line ends with a line feed character (the carriage return is implicit), under DOS every line must end with explicit carriage return and line feed characters.

FTP To use FTP, you need a machine with direct access to the Internet. A sample session is shown below.

```
$ ftp ftp.uu.net
Connected to ftp.uu.net.
220 ftp.UU.NET FTP server
  (Version 6.34 Oct 22 14:32:01 1992) ready.
Name (ftp.uu.net:prefect): anonymous
331 Guest login ok, send e-mail address as
  password.
Password: prefect@guide.com
230 Guest login ok, access restrictions apply.
ftp> cd /published/oreilly/nutshell/maketexwork
250 CWD command successful.
ftp> get README
200 PORT command successful.
150 Opening ASCII mode data connection for
  README (xxxx bytes).
226 Transfer complete.
local: README remote: README
xxxx bytes received in xxx seconds (xxx Kbytes/s)
ftp> binary
200 Type set to I.
(Repeat get commands for the other files.
  They are listed in the README file.)
ftp> quit
221 Goodbye.
$
```

FTPMAIL FTPMAIL is a mail server available to anyone who can send electronic mail to, and receive it from, Internet sites. This includes most workstations that have an email connection to the outside world and CompuServe users. You do not need to be directly on the Internet.

Send mail to *ftpmail@decwrl.dec.com*. In the message body, give the name of the anonymous FTP host and the FTP commands you want to run. The server will run anonymous FTP for you and mail the files back to you. To get a complete help file, send a message with no subject and the single word *help* in the body. The following is an example mail session that should get you the examples. This command sends you a listing of the files in the selected directory and the requested example files. The listing is useful if there's a later version of the examples you're interested in.

```
$ mail ftpmail@decwrl.dec.com
Subject: reply prefect@guide.com
connect ftp.uu.net
chdir /published/oreilly/nutshell/maketexwork
dir
get README
quit
```

A signature at the end of the message is acceptable as long as it appears after *quit*.

BITFTP BITFTP is a mail server for BITNET users. You send it electronic mail messages requesting files, and

it sends you back the files by electronic mail. BITFTP currently serves only users who send it mail from nodes that are directly on BITNET, EARN, or NetNorth. BITFTP is a public service of Princeton University.

To use BITFTP, send mail containing your FTP commands to *BITFTP@PUCC*. For a complete help file, send *HELP* as the message body.

The following is the message body you should send to BITFTP:

```
FTP ftp.uu.net
NETDATA USER anonymous
PASS your Internet e-mail address (not your BIT-
NET address)
CD /published/oreilly/nutshell/maketexwork
DIR
GET README
QUIT
```

Questions about BITFTP can be directed to *MAINT@PUCC* on BITNET.

UUCP UUCP is standard on virtually all UNIX systems and is available for IBM-compatible PCs and Apple Macintoshes. The examples are available by UUCP via modem from UUNET; UUNET's connect-time charges apply. You can get the examples from UUNET whether you have an account or not. If you or your company has an account with UUNET, you will have a system with a direct UUCP connection to UUNET. Find that system, and type (as one line):

```
$ uucp uunet\!~/published/oreilly/
  nutshell/maketexwork/README \
yourhost\!~/yourname
```

The README file should appear some time later (up to a day or more) in the directory */usr/spool/uucppublic/yourname*. If you don't have an account, but would like one so that you can get electronic mail, contact UUNET at 703-204-8000.

If you don't have a UUNET account, you can set up a UUCP connection to UUNET in the United States using the phone number 1-900-468-7727. As of this writing, the cost is 50 cents per minute. The charges will appear on your next telephone bill. The login name is *uucp* with no password. Your entry may vary depending on your UUCP configuration.

Gopher If you are on the Internet, you can use the *gopher* facility to learn about online access to examples through the O'Reilly Online Information Resource. Access *gopher.ora.com* as appropriate from your site.

2.5 Versions of \TeX

The most recent versions of \TeX and METAFONT are version 3.1415 and version 2.71, respectively. Version 3 of \TeX introduced several new features designed to improve support for non-English languages (including the use of

8-bit input and some refinements to hyphenation control). If you use an older version of T_EX, you should upgrade.

Donald Knuth has specified that T_EX's version number converges to π , therefore version 3.1415 is only the fourth minor revision after version 3. The next minor revision will be version 3.14159. Similarly, METAFONT's version number converges to e (2.7182818284. . .).

2.6 Implementations and Platforms

The interface that T_EX presents to the writer is very consistent. Most of the examples described in this book are applicable to every single implementation of T_EX. However, T_EX is not a closed system. It is possible to step outside of T_EX to incorporate special elements into your document or take advantage of the special features of a particular environment. These extensions can dramatically restrict the portability of your documents.

Many of the topics covered in this book offer alternatives in those areas that are less portable. Therefore, it is natural to ask what implementations are really covered.

Before outlining which implementations are covered, let me suggest that this book will be useful even if you are using an implementation not 'officially' covered here. The reality of the situation is this: many, many tools have been ported with T_EX. Many of the tools mentioned in this book are available on platforms that are not specifically discussed. Time and equipment constraints prevented Amiga, Atari, NeXT, VMS, and Windows NT implementations of T_EX from being specifically addressed in this edition of the book.

2.6.1 UNIX

UNIX is probably the most common T_EX platform. The emphasis in this book is on UNIX workstations running X11, producing output for PostScript and HP LaserJet printers.

Linux and other personal computer implementations of UNIX are not addressed specifically; however, with the successful port of X11 to *Linux*, I'm confident that every UNIX tool here can be, or has been, ported to *Linux* (and probably other PC UNIX environments).

The only implementation of the T_EX program for UNIX considered in any detail is the free implementation distributed in *web2c*. This distribution is described in the section called 'Web2C' in Chapter 13, *Non-commercial Environments*. Most of the other UNIX tools discussed here are also free.

2.6.2 MS-DOS

With very few exceptions, the tools in this book are available under MS-DOS. Because PCs are very popular, a lot of effort has gone into porting UNIX tools to MS-DOS. Some packages, however, require a 386SX (or more powerful) processor. For the most part, I focus on PCs running MS-DOS only; however, Microsoft Windows and DesqView are not entirely ignored.

There are quite a few options when it comes to selecting an implementation of the T_EX program under MS-DOS. Several free implementations are discussed as well as some commercial implementations. For more information about these implementations, consult Chapter 13, *Non-commercial Environments*, and Chapter 14, *Commercial Environments*.

2.6.3 OS/2

In this book, OS/2 is treated primarily as a superset of MS-DOS. When possible, I look at OS/2-specific versions of each utility, but rely on MS-DOS as a fall-back.

Extensions to emT_EX for OS/2 are explored, as are editing environments such as *epm*. The multi-threaded nature of OS/2 allows more complete porting of UNIX tools. When better ports are available for OS/2, they are discussed.

2.6.4 Macintosh

The Macintosh is very different from the systems described above. Chapter 15, *T_EX on the Macintosh*, discusses the Macintosh environment in detail.

There are four implementations of T_EX for the Macintosh. Three are freely available, and one is commercial: CMacT_EX is free, OzT_EX and DirectT_EX are shareware, and *Textures* is a commercial package from Blue Sky Research.

2.7 We'd Like to Hear From You

We have tested and verified all of the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing:

O'Reilly & Associates, Inc.
103 Morris Street, Suite A
Sebastopol, CA 95472
1-800-998-9938 (in the US or Canada)
1-707-829-0515 (international/local)
1-707-829-0104 (FAX)

You can also send us messages electronically. To be put on the mailing list or request a catalog, send email to:

nuts@ora.com (via the Internet)
uunet!ora!nuts (via UUCP)

To ask technical questions or comment on the book, send email to:

bookquestions@ora.com (via the Internet)

2.8 Acknowledgments

This book would not exist if I had not received support and encouragement from my friends and colleagues, near and far. I owe the deepest debt of gratitude to my wife, Deborah, for patience, understanding, and support as I progressed through what is easily the most all-consuming task I have ever undertaken.

The earliest draft of this book came about because my advisor at the University of Massachusetts, Eliot Moss, allowed me to tinker with the \TeX installation in the Object Systems Lab and was always able to suggest ways to make it better. My friends and colleagues at UMass, Amer Diwan, Darko Stefanović, Dave Yates, Eric Brown, Erich Nahum, Jody Daniels, Joe McCarthy, Ken Magnani, Rick Hudson, and Tony Hosking, asked all the hard questions and didn't seem to mind when I used them as guinea pigs for my latest idea.

I'm indebted also to Eberhard Mattes, Geoffrey Tobin George D. Greenwade, Peter Schmitt, Sebastian Rahtz, and Tomas Rokicki, who provided technical review comments on the materials presented here. Jim Breen and Ken Lunde offered invaluable feedback on Chapter 7.

And I'd like to thank a lot of people at O'Reilly for their help and enthusiasm; in particular, my editor, Debby Russell, offered advice, helpful criticism, and support beyond the call of duty (Debby keyed most of the index for this book as production deadlines drew near and other arrangements fell through); Chris Tong organized the raw entries into a usable index; Lenny Muellner, Donna Woonteiler, and Sheryl Avruch allowed me to work on the book when it wasn't technically my job; Stephen Spainhour copyedited it into English with the help of Leslie Chalmers and Kismet McDonough (Stephen offered helpful suggestions along the way, too); Jennifer Niederst helped me get the design right; and Chris Reilly created the figures and screen dumps. I enjoyed working with everyone at O'Reilly so much that I left UMass and joined the production department myself ;-).

Several companies provided review copies of their software while I was writing this book. I would like to thank ArborText, Blue Sky Research, Borland International, The Kinch Computer Company, LaserGo, Personal \TeX , TCI Software Research, and Y&Y, for their generosity.

Finally, I'd like to thank the entire Internet \TeX community. Countless thousands of questions and answers on the Net refined my understanding of how \TeX works and what it can do.

— * —

3 Chapter 4 from book: Macro Packages

Everyone who uses \TeX uses a macro package (also called a 'format'). A macro package extends \TeX to provide functionality that is suited to a particular task or set of tasks.

This chapter provides a summary of \TeX macro packages. General-purpose packages designed to typeset a wide range of documents—articles, books, letters, and reports—are examined first. The general-purpose packages described are Plain \TeX , Extended Plain \TeX , $\LaTeX 2_{\epsilon}$, \LaTeX , \AMSTeX , $\AMSI\LaTeX$, Lollipop, and \TeXinfo . After surveying the general-purpose packages, several special-purpose packages designed to handle specific tasks—

typesetting transparencies, music, chemistry or physics diagrams, etc.—are described. The special-purpose packages surveyed are $\text{Sl}\TeX$, $\text{Foil}\TeX$, Seminar , $\text{Music}\TeX$, ChemStruct , and $\text{Chem}\TeX$.

There are a lot of overlapping features and similar commands in the general-purpose packages. To understand why this is the case, consider how a new macro package comes into existence. An ambitious person, who is very familiar with \TeX , decides that there are some things she would like to express in her documents that are difficult to express with existing formats. Perhaps, for example, no existing format produces documents that match the precise specifications required for publishing in her field, or perhaps she has in mind a whole new document structuring paradigm. A more mundane possibility is simply that she has been customizing an existing format for some time and now feels it has enough unique features to be useful to others.

In any event, a new format is born. Now, if this format is designed for a very specific task, writing multiple-choice mathematics exams, for example, it might not have very many general-purpose writing features. On the other hand, if it is designed for writing longer, more general documents (e.g., history textbooks or papers to appear in a particular journal) then there are a number of features that it is likely to include; provisions for numbered lists, cross references, tables of contents, indexes, and quotations are all examples of features common to many documents.

To support these common features, many macro packages have similar control sequences. This stems from the fact that they are all built on top of a common set of primitives and that macro package authors tend to copy some features of other packages into their own.

You may find that you'd like to use the features of several different packages in the same document. Unfortunately, there is no provision for using multiple formats to process a single document. The features required to process most documents are shared by all of the general-purpose formats, however. You are more likely to need multiple macro packages if you want to use a special format to construct a diagram or figure and incorporate it into a document. Chapter 6, *Pictures and Figures*, describes several ways to take 'electronic snippets' of one document and insert them into another, which is one possible solution to this problem.

If you're beginning to feel a little lost, have no fear. Most general-purpose formats are sufficient for most documents. And there's no reason why every document you write has to be done with the same format. Many people find \LaTeX and Plain \TeX sufficient, but if you're writing an article for a particular journal and someone has written a format specifically for that journal's documents, by all means use it. It is more likely, however, that someone has written a style file which tailors \LaTeX to the requirements of the journal.

In addition to describing some common macro packages in this chapter, I'll describe how to build format files for them. If the packages that you want to use have already been installed at your site, you can ignore the installation sections.

The packages that you find most convenient will depend on the tasks you perform and how well each package suits your work style. The list of packages in this chapter is not meant to be all-inclusive, nor is it my intention to suggest which packages are best. Use the ones *you* like, for whatever reasons.

I can hear some of you already, 'I don't really need a macro package,' you say, 'I can roll my own with just T_EX.'

And you are absolutely correct.

I don't recommend it, however. It's akin to using your compiler without any of the built-in functions. Most T_EX primitives offer little support by themselves for writing documents.

A New Perspective: L^AT_EX_{2 ϵ} (and its successor, which will be L^AT_EX₃) seek to address many of the problems mentioned above by defining a core L^AT_EX format with extension packages to provide custom features.

The L^AT_EX_{2 ϵ} system provides a single format file that supports L^AT_EX, AMST_EX, and SliT_EX. For the time being, L^AT_EX_{2 ϵ} is described separately in this chapter along with the other formats. Be aware, however, that L^AT_EX_{2 ϵ} is now the standard L^AT_EX² and you should migrate to it as soon as possible.

3.1 Installation: Making Format Files

A format file, as described in Chapter 3, *Running T_EX*, is a special 'compiled' version of a macro package. The iniT_EX program interprets all of the control sequences in a macro package and writes the corresponding memory image into a file. Loading a format file is much faster than loading individual macro packages in your document because T_EX does not have to interpret any of the control sequences while it is also processing text.

In general, all format files should be stored in the same directory.⁴ If you install T_EX in a directory called *tex*, then formats typically go in a directory called *formats* in the *tex* directory. This is not universally the case because you need separate format directories for big and small T_EXs.⁵

Usually, an environment variable indicates where the format files are located. Environment variables are a common way of customizing your interaction with programs. They are usually set in your *AUTOEXEC.BAT* file for MS-DOS, *CONFIG.SYS* for OS/2, or the *rc*-file for a shell (i.e., *.cshrc*, *.kshrc*) on UNIX systems. Any good reference book

for your operating system or shell will describe how to use environment variables.

A common name for the environment variable that indicates where T_EX formats are located is `TEXFORMATS`. Implementations that provide big and small T_EXs need another variable to indicate the directory that contains formats for big T_EX.

IniT_EX is not always a separate program. Some implementations of T_EX combine the functionality of T_EX and iniT_EX into a single program and use a special switch at runtime to determine which function to perform. In this chapter, all of the examples use the program name *initex* to identify iniT_EX. If you use an implementation that doesn't provide a separate iniT_EX program, you should use the T_EX program with the iniT_EX switch instead. For example, for emT_EX, use *tex /i* instead of *initex* when you build a format.

Like T_EX, iniT_EX needs to be able to find input files. Usually, this is accomplished by searching the directories listed in the `TEXINPUTS` environment variable. Place the input files that iniT_EX needs in a directory on the `TEXINPUTS` path before running iniT_EX unless otherwise directed. The `TEXINPUTS` environment variable is discussed in the 'User Files' section of Chapter 3.

Hyphenation Patterns

In order for T_EX to correctly hyphenate words, every format file must contain a set of hyphenation patterns. The patterns are part of an algorithmic solution to the problem of breaking a word into syllables for hyphenation.

The details of the hyphenation algorithm (given in Appendix H of *The T_EXbook*) are too complex to describe here, but two aspects of this solution deserve particular emphasis. First, using patterns means that a dictionary of hyphenated words is not necessary.⁶ This saves a lot of space and time. Second, by loading different sets of patterns, T_EX can achieve equal success at hyphenating any language—even English ;-). There are actually at least two sets of hyphenation patterns for English, one for British English and one for American English. Chapter 7, *International Considerations*, describes how to load multiple sets of hyphenation patterns for typesetting multilingual documents.

3.2 General-purpose Formats

This section describes several macro packages that are designed for formatting standard documents like articles or books. In order to provide some form of comparative measure, each macro package is used to create the same document, a one-page report that looks like Figure 1. I constructed this example to demonstrate a few common elements in a document: several sizes of headings, a paragraph of text, inline and displayed mathematics, and a few

²At the time of this writing, it's actually still in test-release, but it may be available as a standard release by the time you read this.

⁴On Macintosh systems and other environments that don't have directories, format files are typically stored in their own folders (or other metaphorically appropriate place ;-).

⁵The distinction between big and small T_EX is described in the section called 'What Do You Run?' in Chapter 3.

⁶A small set of exceptions is maintained because the algorithm isn't perfect.

fonts. There are lots of other things that aren't shown (tables, figures, footnotes, etc.), and these elements vary as much as any other in the different macro packages.

Observant readers will notice that the examples are shown in the Computer Modern fonts while the rest of this book uses different fonts.⁷ There are a number of reasons why the example is shown in Computer Modern. For one thing, all of the formats discussed here use the Computer Modern fonts by default. Using a different set of fonts would only add more complexity to each example. A more subtle problem is that I do not have appropriate mathematics fonts for Garamond. There are a number of complex issues involving the use of fonts in T_EX. They are discussed in Chapter 5, *Fonts*.

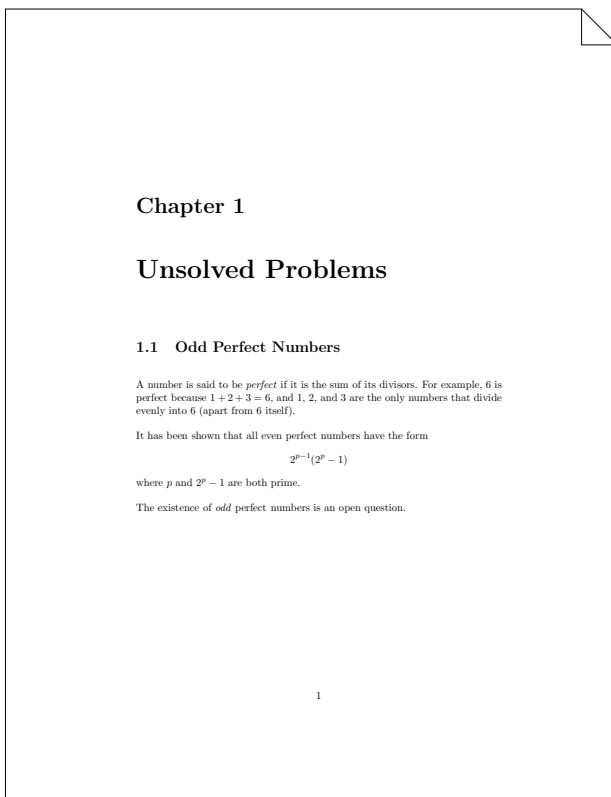


Figure 1: Sample page

3.2.1 Plain T_EX

Plain T_EX is the format written by Donald Knuth during the development of T_EX. It is described fully in *The T_EXbook*.

Plain T_EX ties together the T_EX primitives in a way that makes it practical to work in T_EX. If you do not have a computer programming background, you may find Plain T_EX a little bit intimidating. It is definitely a 'roll your own' environment. Although it demonstrably contains all of the functionality required to write everything from letters to books, there is very little 'user-friendly' packaging around the internals of T_EX.

⁷Really observant readers may have noticed that it's a version of Garamond ;-)

Aside from user interface considerations, which are highly subjective, Plain T_EX lacks some functionality when compared to other formats. There is no provision in Plain T_EX for automatically numbered sections, labelled figures, tables of contents, indexes, or bibliographies. Any of these functions can be constructed in Plain T_EX if you are willing to invest the time and energy required to write your own macros, but they are not built into Plain T_EX.

If you enjoy writing your own macros or plan to produce novel types of documents, a firm grasp of Plain T_EX will allow you to write anything in T_EX. A firm grasp of Plain T_EX also makes it easier to understand and modify other formats (like L^AT_EX) that are built on top of Plain T_EX.

In addition, Plain T_EX is the only format that is always distributed with T_EX. The other formats discussed in this chapter are freely available but do not come with T_EX.

The Plain T_EX input that produces the report in Figure 1 is shown in Example 1.

Example 1: Plain T_EX Input

```
% Format: Plain
\font\chapfont=cmbx12 scaled 1728
\font\titlefont=cmbx12 scaled 2073
\font\secfont=cmbx12 scaled 1200

\parskip=\baselineskip
\parindent=0pt
\hsize=5in
\hoffset=.75in

\leftline{\chapfont Chapter 1}
\vskip36pt

\leftline{\titlefont Unsolved Problems}
\vskip36pt

\leftline{\secfont 1.1\ \ Odd Perfect Numbers}
\vskip12pt

A number is said to be perfect if it is the sum of its divisors. For example, $6$ is perfect because $1+2+3 = 6$, and $1$, $2$, and $3$ are the only numbers that divide evenly into $6$ (apart from $6$ itself).

It has been shown that all even perfect numbers have the form  $2^{p-1}(2^p-1)$  where $p$ and $2^p-1$ are both prime.

The existence of odd perfect numbers is an open question.

\bye
```

Building the Plain T_EX format To build Plain T_EX, you need only two files: *plain.tex* and *hyphen.tex*. These files are distributed with T_EX so they should be available as soon as you have installed T_EX. The *hyphen.tex* file is language-dependent. Readers who frequently work with non-English text should read Chapter 7 for more information about obtaining non-English hyphenation patterns.

The command:

```
$ initex plain \dump
```

will create the Plain T_EX format. Move the resulting files, *plain.fmt* and *plain.log*, into your T_EX formats directory.

3.2.2 Extended Plain T_EX

Extended Plain T_EX extends Plain T_EX in a number of useful ways without forcing you to use any particular ‘style’ of output. The argument is this: although Plain T_EX really doesn’t provide all of the features that you need (tables of contents, cross references, citations, enumerated lists, convenient access to verbatim input, etc), many of these features don’t have any direct impact on the appearance of your document. Unfortunately, other general-purpose macro packages like L^AT_EX, which do provide these features, tend to force you to accept their notion of what the typeset page should look like.⁸

Extended Plain T_EX is an attempt to solve that problem. It provides many behind-the-scenes features without providing any general page layout commands (like `\chapter` or `\section`), which means that these features can be used inside Plain T_EX without much difficulty and without changing the layout of typeset pages.

Building the Extended Plain T_EX format To build the Extended Plain T_EX format, you need the *plain.tex* and *hyphen.tex* files required to build the Plain format as well as the *eplain.tex* file distributed with Extended Plain T_EX.⁹ Make and install the Plain T_EX format first, then change to the directory that contains the Extended Plain T_EX distribution.

The command:

```
$ initex &plain eplain \dump
```

will create the Extended Plain T_EX format. Move the resulting files, *eplain.fmt* and *eplain.log*, into your T_EX formats directory.

3.2.3 L^AT_EX_{2 ϵ} Versus L^AT_EX

The tremendous popularity of L^AT_EX in the T_EX community has had an unfortunate side effect: because it is a very familiar and flexible format, many people have used it as the basis for extensions of one sort or another. This has resulted in a wide range of (slightly) incompatible formats and a lot of frustration.

This situation is being rectified by a new release of L^AT_EX, currently called L^AT_EX_{2 ϵ} . The new release replaces the existing dialects of L^AT_EX (L^AT_EX with and without NFSS, SliT_EX, AMSL^AT_EX, etc.) with a single core system and a set of extension packages. L^AT_EX_{2 ϵ} includes a compatibility mode which will allow it to continue to format existing documents without change (provided that they do not rely on local modifications to the L^AT_EX format, of course). Local modifications can also be incorporated

into the L^AT_EX_{2 ϵ} system as extension packages, making L^AT_EX_{2 ϵ} a complete replacement for all existing versions of L^AT_EX and packages closely derived from L^AT_EX.

The most significant and least compatible difference between L^AT_EX and L^AT_EX_{2 ϵ} is the font selection scheme. There are many control sequences for selecting fonts in L^AT_EX. Some control the typeface (`\rm`, `\tt`, `\sc`, etc.); some the size (`\small`, `\normalsize`, `\large`, etc.); and some the appearance (`\it`, `\bf`, `\em`, etc.).

Under the Old Font Selection Scheme (OFSS), the control sequences for selecting a font completely override any font selection already in place. Consider, for example, the control sequences `\it` and `\bf`, which switch to italic and boldface. Using `\bf\it` produces italic text, and `\it\bf` produces boldface text, and *neither* produces boldfaced-italic text (which is probably what you wanted).

Under the New Font Selection Scheme, typeface (called *family* in NFSS parlance), appearance (called *series* and *shape*), and size are viewed as orthogonal components in font selection. Because these parameters are independent, selecting an italic appearance with the `\it` control sequence switches to italic in the current typeface and size. Under the NFSS, `\bf\it` *does* select boldface-italic in both the current typeface and size (if it is available).

L^AT_EX_{2 ϵ} supports only the NFSS, version 2 (called NFSS2). For more than a year, the NFSS (initially version 1, and more recently version 2) has been available as an extension for L^AT_EX 2.09. However, in light of stable test releases of L^AT_EX_{2 ϵ} , the NFSS2 package for L^AT_EX 2.09 has been withdrawn. The discussion of NFSS in this book applies equally well to L^AT_EX 2.09 with NFSS2, but it is described in terms of L^AT_EX_{2 ϵ} in an effort to be more applicable in the future. The NFSS is discussed in more detail in Chapter 5.

3.2.4 L^AT_EX_{2 ϵ}

Leslie Lamport’s L^AT_EX format is probably the most commonly used T_EX format. It is described in *L^AT_EX: A Document Preparation System* and many other T_EX books. L^AT_EX_{2 ϵ} is the new standard L^AT_EX. It is described in *The L^AT_EX Companion*. The next edition of *L^AT_EX: A Document Preparation System* will also describe L^AT_EX_{2 ϵ} .

At the time of this writing, L^AT_EX_{2 ϵ} is available only in a test release, but by the time you read this, it is likely to be available as the new standard L^AT_EX. It is described first in this chapter to emphasize that you should begin using L^AT_EX_{2 ϵ} as soon as possible. Once L^AT_EX_{2 ϵ} is out of testing, it will become L^AT_EX, and support for older versions will not be provided (at least not by the L^AT_EX developers). L^AT_EX_{2 ϵ} includes a compatibility mode for old L^AT_EX documents so the transition should be relatively painless.

⁸Of course, that’s not strictly true. You can change the page layout of L^AT_EX (and most other packages) to be almost anything, but it does require learning a lot about the macro package. If you are already familiar with Plain T_EX (or some other Plain T_EX-derived package), you probably have a set of macros that produce documents in the style you like. Why reinvent the wheel?

⁹Extended Plain T_EX is available in the *macros/eplain* directory in the CTAN archives.

The central theme of L^AT_EX is ‘structured document preparation.’ An ideal L^AT_EX document is described entirely in terms of its structure: chapters, sections, paragraphs, numbered lists, bulleted items, tables, figures, and all the other elements of a document are identified descriptively. For example, you enclose figures in a figure *environment* identified by the control sequences `\begin{figure}` and `\end{figure}`.

When you are ready to print your document, select an appropriate document class, and L^AT_EX formats your document according to the rules of the selected style. In the case of the ideal document, it might first be printed in a magazine or newsletter using the article class. Later, when it is incorporated into a book, selecting the book class is *all* that is required to produce appropriate output; the document itself is unchanged.

L^AT_EX is written on top of Plain T_EX. This means that almost any control sequence or macro that you learn about in Plain T_EX can also be used in L^AT_EX. Of course, L^AT_EX insulates you from many Plain T_EX commands by wrapping a much more user-friendly interface around them.

The L^AT_EX_{2 ϵ} input that produces the sample page in Figure 1 is shown in Example 2. The only difference between this document and an old L^AT_EX document is the use of the `\documentclass` declaration instead of the `\documentstyle` declaration.¹⁰ For more complex documents, other minor changes may also be necessary.

Example 2: L^AT_EX_{2 ϵ} Input

```
% Format: LaTeX2e
\documentclass{report}

\setlength{\parskip}{\baselineskip}
\setlength{\parindent}{0pt}

\begin{document}

\chapter{Unsolved Problems}

\section{Odd Perfect Numbers}

A number is said to be \emph{perfect} if it
is the sum of its divisors. For example,
$6$ is perfect because  $(1+2+3 = 6)$ , and
$1$, $2$, and $3$ are the only numbers that
divide evenly into $6$ (apart from 6 itself).

It has been shown that all even perfect numbers
have the form  $(2^{p-1})(2^p-1)$  where $p$
and  $(2^p-1)$  are both prime.

The existence of \emph{odd} perfect numbers is
an open question.
\end{document}
```

Building the L^AT_EX_{2 ϵ} format The L^AT_EX_{2 ϵ} distribution is available from the directory `macros/latex2e/core` in the CTAN archives.

The following steps will build the L^AT_EX_{2 ϵ} format. For more complete installation instructions, read the file `install.l2e` in the L^AT_EX_{2 ϵ} distribution.

1. Place the L^AT_EX_{2 ϵ} distribution in a temporary directory and make that directory the current directory. After the installation is complete, you will need to move only selected files into the standard places.
2. Restrict to only the current directory the directories that T_EX searches for input files
This can usually be accomplished by setting the environment variable `TEXINPUTS`¹¹ to a single period or the absolute path name of the current directory.
3. Copy the `hyphen.tex` file from the Plain T_EX distribution into the current directory.
4. Issue the command:
`$ initex unpack2e.ins`
This will unpack all of the distribution files.
5. Build the format file by issuing the command:
`$ initex latex2e.ltx`
Move the resulting files `latex2e.fmt` and `latex2e.log` into the T_EX formats directory.
6. In addition to the files needed to build the format, unpacking the L^AT_EX_{2 ϵ} distribution creates many files that are needed for formatting documents. These files must be placed in a location where T_EX will find them. However, in order to maintain a functioning L^AT_EX 2.09 system, you must not place the new files in the same input directory as the existing files.¹²

Create a new directory (or folder) for the new files. On the UNIX system that I use, where existing input files are stored in a directory called `/usr/local/lib/tex/inputs`, I created `/usr/local/lib/tex/latex2e` to store the new files. You will have to add the new directory to the *front* of the list of directories that T_EX searches for input files whenever you format a document with L^AT_EX_{2 ϵ} .

Move the files that the installation script produces into the new directory. Move the files `docstrip.tex`, `latexbug.tex`, `sfontdef.ltx`, `slides.ltx`, `testpage.tex`, and all of the files that end in `.cfg`, `.cls`, `.clo`, `.def`, `.fd`, and `.sty`. You should also move the files `glo.ist` and `gind.ist` someplace where `MakeIndex` can find them. (`MakeIndex` is described in Chapter 12 *Bibliographies, Indexes, and Glossaries*.)

One of the aspects of the test releases that continues to change is the exact list of files that must be moved. Consult the `install.l2e` file in the distribution for the exact list. The list above is from the test version of January 28, 1994.

3.2.5 L^AT_EX

This section briefly covers L^AT_EX version 2.09. This version of L^AT_EX is still very widely used but it is being phased out.

¹⁰L^AT_EX_{2 ϵ} will process documents that use `\documentstyle` in L^AT_EX 2.09 compatibility mode.

¹¹Under emT_EX, this variable is called `TEXINPUT`. On the Macintosh, file searching is frequently controlled by a configuration file or dialog box.

¹²Strictly speaking, this is only true for files that have the extension `.sty` because the old version of L^AT_EX will not attempt to use the other files.

The L^AT_EX input to produce the sample page in Figure 1 is shown in Example 3.

Example 3: L^AT_EX Input File

```
% Format: LaTeX
\documentstyle{report}

\setlength{\parskip}{\baselineskip}
\setlength{\parindent}{0pt}

\begin{document}

\chapter{Unsolved Problems}

\section{Odd Perfect Numbers}

A number is said to be {\em perfect\} if it
is the sum of its divisors. For example,
$6$ is perfect because  $(1+2+3 = 6)$ , and $1$,
$2$, and $3$ are the only numbers that divide
evenly into $6$ (apart from 6 itself).
      %<TeX_Marker>

It has been shown that all even perfect numbers
have the form  $(2^{p-1})(2^p-1)$  where $p$
and  $(2^p-1)$  are both prime.

The existence of {\em odd\} perfect numbers is
an open question.
\end{document}
```

Support for the NFSS in L^AT_EX 2.09 has been withdrawn. If you need to build a format with support for NFSS, consult the ‘L^AT_EX2_ε’ section of this chapter.

Building the L^AT_EX format with the OFSS The L^AT_EX distribution¹³ includes three subdirectories, *sty*, *doc*, and *general*. All of the L^AT_EX files required to build the format file are in the *general* subdirectory. You will also need the *hyphen.tex* file required to build the Plain format.

In the *general* subdirectory, the command:

```
$ initex lplain
```

will create the L^AT_EX format. Move the resulting files, *lplain.fmt* and *lplain.log*, to the T_EX formats directory. In order to complete the installation, copy the files from the *sty* directory in the L^AT_EX distribution into a directory where T_EX searches for input files.

3.2.6 AMST_EX

When the American Mathematical Society selected T_EX as a document preparation system, they decided to extend it in a number of ways to make the creation of papers and journals easier. They had two goals: to make it easier for authors to write mathematical papers in T_EX and to make the resulting papers conform to a particular set of style specifications. AMST_EX is described completely in *The Joy of T_EX*.

AMST_EX provides many commands that resemble L^AT_EX environments. These have the form `\environment ...`

`\endenvironment`. In addition, AMST_EX provides the notion of a document style to control style-related formatting issues.

Another important contribution made by the American Mathematical Society when creating AMST_EX was the construction of a large number of new fonts. The American Mathematical Society provides fonts with many more mathematical symbols than the fonts that come with T_EX. These fonts are available as a separate package and can be used with any T_EX macro package, not just AMST_EX.

The AMST_EX input required to produce the document in Figure 1 is shown in Example 4. The result of formatting this document does not appear exactly like Figure 1 because AMST_EX uses the style conventions of the American Mathematical Society.

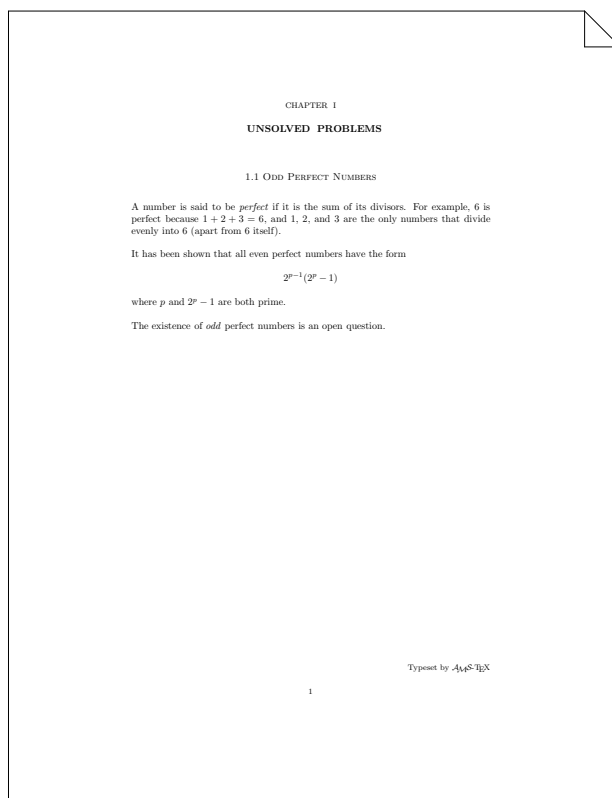


Figure 2: AMS sample page

Example 4: AMST_EX Input File

```
% Format: AMSTeX
\documentstyle{amspt}

\parindent=0pt
\parskip=\baselineskip
\hoffset=.75in

\topmatter
\title \chapter{1} Unsolved Problems\endtitle
\endtopmatter

\document
```

¹³L^AT_EX is available in the *macros/latex/distrib/latex* directory in the CTAN archives.

```
\head{1.1} Odd Perfect Numbers\endhead
```

A number is said to be `{\it perfect\}` if it is the sum of its divisors. For example, `6` is perfect because $1+2+3 = 6$, and `1`, `2`, and `3` are the only numbers that divide evenly into `6` (apart from `6` itself).

It has been shown that all even perfect numbers have the form $2^{p-1}(2^p-1)$ where p and 2^p-1 are both prime.

The existence of `{\it odd\}` perfect numbers is an open question.
`\enddocument`

Building the AMST_EX format In order to build the AMST_EX format, you need the *plain.tex* and *hyphen.tex* files from Plain T_EX and the *amstex.ini* and *amstex.tex* files from the AMST_EX distribution.

The command:

```
$ initex amstex.ini
```

will create the AMST_EX format. Move the resulting files, *amstex.fmt* and *amstex.log*, into your T_EX formats directory.

3.2.7 AMSI_AT_EX

AMSI_AT_EX, like AMST_EX, provides many features to make typesetting mathematics convenient while meeting the standards of the American Mathematical Society for publication. However, AMST_EX lacks many of the features that are present in L_AT_EX, like automatically numbered sections and tools for creating tables of contents and indexes.

When L_AT_EX gained popularity, many authors requested permission to submit articles to the American Mathematical Society in L_AT_EX. In 1987, the American Mathematical Society began a project to combine the features of AMST_EX with the features of L_AT_EX. The result is AMSI_AT_EX.

AMSI_AT_EX provides all of the functionality of L_AT_EX because it is an extension of L_AT_EX. It also provides the functionality of AMST_EX in L_AT_EX syntax and access to additional mathematical constructs and math symbols not present in L_AT_EX.

The input required to produce Figure 1 is not shown because they do not differ significantly from the L_AT_EX sample. Because the sample document doesn't use any of AMSI_AT_EX's additional features, it is exactly the same as the L_AT_EX document.

Building the AMSI_AT_EX format The AMSI_AT_EX macros are embodied entirely in style files for L_AT_EX. It is not necessary to build a special format file. However, the AMSI_AT_EX macros require the New Font Selection Scheme (NFSS). Consult the section on L_AT_EX, above, for instructions on building the L_AT_EX format with NFSS.

3.2.8 Lollipop

It can be argued that L_AT_EX has the following deficiency: although there are many different style options available, it

is not easy for a novice user to change a style option. Changing the internals of most L_AT_EX style options requires a deep understanding of T_EX.

The Lollipop format is very different from the other formats. The central thrust of Lollipop is that it should be easy to change and customize document styles. All Lollipop documents are built from five different generic constructs: headings, lists, text blocks, page grids, and external items.

The Lollipop input required to produce a document like the sample page in Figure 1 is shown in Example 5.

Example 5: Lollipop Input File

```
% Format: Lollipop
\DefineHeading:Chapter
  breakbefore:yes whiteafter:12pt
  line:start PointSize:20 Style:bold
    literal:Chapter Spaces:1 ChapterCounter
  line:stop
  vwhite:36pt
  line:start PointSize:24 Style:bold title
  line:stop
  vwhite:24pt
  Stop
\DefineHeading:Section
  whitebefore:{20pt plus 2pt} whiteafter:14pt
  line:start PointSize:14 Style:bold
    ChapterCounter . SectionCounter
  Spaces:1 title line:stop
  label:start ChapterCounter . SectionCounter
  label:stop
  Stop
\GoverningCounter:Section=Chapter
\AlwaysIndent:no
\Distance:parskip=12pt
\Distance:hoffset=-.75in
\Distance:voffset=-.5in
\Start
\Chapter Unsolved Problems

\Section Odd Perfect Numbers
```

A number is said to be `{\it perfect\}` if it is the sum of its divisors. For example, `6` is perfect because $1+2+3 = 6$, and `1`, `2`, and `3` are the only numbers that divide evenly into `6` (apart from `6` itself).

It has been shown that all even perfect numbers have the form $2^{p-1}(2^p-1)$ where p and 2^p-1 are both prime.

The existence of `{\it odd\}` perfect numbers is an open question.
`\Stop`

Building the Lollipop format To make the Lollipop format, you need the Lollipop distribution and the file *hyphen.tex* from the Plain T_EX distribution.

This command will create the Lollipop format:

```
$ initex lollipop \dump
```

It should be performed in the directory where you installed the Lollipop distribution so that all of the Lollipop files can be located. Move the resulting files *lollipop.fmt* and *lollipop.log*, into your T_EX formats directory.

3.2.9 T_EXinfo

The T_EXinfo format is a general-purpose format, but it was designed to support a particular application: to produce both online documentation and professional quality typeset documentation from the same source file. It is discussed in more detail in Chapter 10 *Online Documentation*.

The input file shown in Example 6 produces the typeset output shown in Figure 3. The input file for this example is complicated by the fact that it contains mathematics. None of T_EX's sophisticated mechanisms for handling mathematics are applicable to plain ASCII online documentation. The online documentation produced by the example in Example 6 is shown in Figure 4.

The T_EXinfo format is the official documentation format of the Free Software Foundation (FSF). Although less commonly used, a L^AT_EX variant called L^AT_EXinfo is also available.

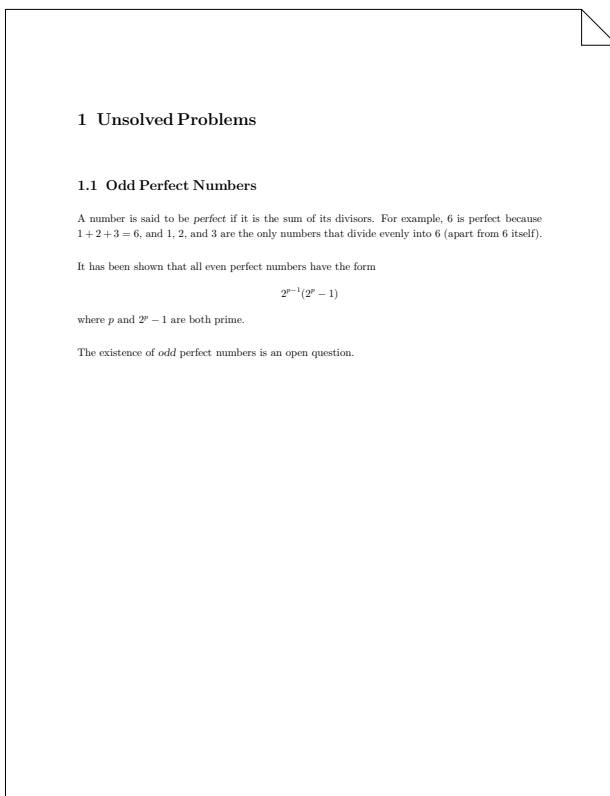


Figure 3: T_EXinfo sample page

Example 6: T_EXinfo Input

```
\input texinfo @c -*- TeXinfo -*-
@setfilename perf-inf.inf
@ifinfo
  @paragraphindent 0
@end ifinfo
@iftex
  @defaultparindent=0pt @parindent=0pt
@end iftex

@node Top, , (dir), (dir)
@chapter Unsolved Problems
@section Odd Perfect Numbers
```

A number is said to be @i{perfect} if it is the sum of its divisors. For example, 6 is perfect because

```
@tex $1+2+3 = 6$,
@end tex
```

```
@ifinfo
  1+2+3 = 6,
@end ifinfo
and 1, 2, and 3 are the only numbers that divide evenly into 6 (apart from 6 itself).
```

It has been shown that all even perfect numbers have the form

```
@tex $$2^{p-1}(2^p-1)$$ where $p$ and $2^{p-1}$
```

```
@end tex
@ifinfo
@center 2^(p-1) (2^p - 1)
```

```
where p and 2^p - 1
@end ifinfo
are both prime.
```

```
The existence of @i{odd} perfect numbers is an open question.
@bye
```

```
Unsolved Problems
*****
```

```
Odd Perfect Numbers
=====
```

A number is said to be perfect if it is the sum of its divisors. For example, 6 is perfect because 1+2+3 = 6, and 1, 2, and 3 are the only numbers that divide evenly into 6 (apart from 6 itself).

It has been shown that all even perfect numbers have the form

$$2^{(p-1)} (2^p - 1)$$

where p and 2^p - 1 are both prime.

The existence of odd perfect numbers is an open question.

Figure 4: Online documentation produced by MakeInfo

3.2.10 Other Formats

There are a number of other macro packages available for T_EX. Some of them are summarized below. The fact that they are not discussed more fully here (or listed below, for that matter) is not intended to reflect on the quality of the format. The formats discussed above are examples of the ways in which T_EX can be extended. All of the formats below extend T_EX in a way similar to one of the formats already mentioned. For any particular application, one of these macro packages might be a better choice than the formats discussed above.

EDMAC Provides support for typesetting critical editions of texts in a format similar to the Oxford Classical Texts with marginal line numbers and multiple series of footnotes and endnotes keyed by line number. EDMAC is available from the CTAN archives in the directory *macros/plain/contrib/edmac*.

INRST \TeX Provides support for multilingual documents in French and English. INRST \TeX is available from the CTAN archives in the directory *macros/inrstex*.

LamST \TeX Extends AMST \TeX with L \AA TeX-like features and improved support for commutative diagrams. LamST \TeX is available from the CTAN archives in *macros/lamstex*.

REVTeX Extends L \AA TeX to provide support for typesetting articles for journals of the American Physical Society, the Optical Society of America, and the American Institute for Physics. REVTeX is available in the directory *macros/latex/contrib/revtex* of the CTAN archives.

TeXsis Provides facilities for typesetting articles, papers, and theses. It is particularly tuned for physics papers. TeXsis also provides support for other kinds of documents, such as letters and memos. It is based upon Plain TeX. TeXsis is available from the CTAN archives in the directory *macros/texsis*.

In addition to REVTeX and TeXsis, there are several other packages in the *macros* directory on CTAN that were designed for typesetting documents about physics: PHYSE, PHYZZX, and PSIZZLE.

TeX/Mathematica Supports interactive use of Mathematica on UNIX workstations running GNU emacs. Mathematica explorations can be annotated with TeX/L \AA TeX, and Mathematica graphics can be incorporated into documents. TeX/Mathematica is available from the CTAN archives in the directory *macros/mathematica*.

ScriptTeX Supports typesetting screenplays in TeX. ScriptTeXmacro packages!ScriptTeX is available from the CTAN archives in the directory *macros/scripttex*.

VerTeX Supports typesetting articles for economic journals. VerTeX is available from the CTAN archives in the directory *macros/plain/contrib/vertex*.

3.3 Special-purpose Formats

In addition to the general-purpose packages discussed above, there are dozens, if not hundreds, of extensions to TeX that are designed for very specific tasks. Many of the extensions are L \AA TeX style files; they provide styles for many academic journals, university theses, resumés, diagrams of various sorts, PostScript interfaces, linguistics, multinational language support, UNIX ‘man’ pages, program listings, and countless other tasks.

To give you a feel for the range of tasks that TeX can perform, I’ve selected a few packages to highlight the latitude of customization that is possible. Figure 5 shows the chemical structure of caffeine (a molecule dear to my heart) rendered with the ChemTeX package. The source is shown in Example 7. Another chemistry package, ChemStruct, was used to draw Figure 6. Its source is shown in Example 8. Taking TeX in another direction, the MusicTeX package was used to typeset the first two bars of Mozart’s K545 sonata in C-major in Figure 7. The MusicTeX input is shown in Example 9. Several more examples are presen-

ted in Chapter 7 where formats for typesetting non-English languages are described.

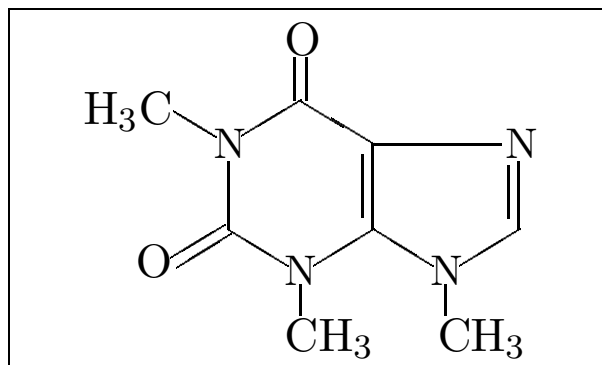


Figure 5: Caffeine by ChemTeX

Example 7: The ChemTeX Source for Caffeine

```
\initial
\len=4
\def\H{\hbox{\rm H}}
\def\C{\hbox{\rm C}}
\def\O{\hbox{\rm O}}

\l \purine{\$H_3\C\$}{\$O\$}{\$C\H_3\$}
{Q}{\$O\$}{Q}{Q}{D}{\$C\H_3\$} \l
```

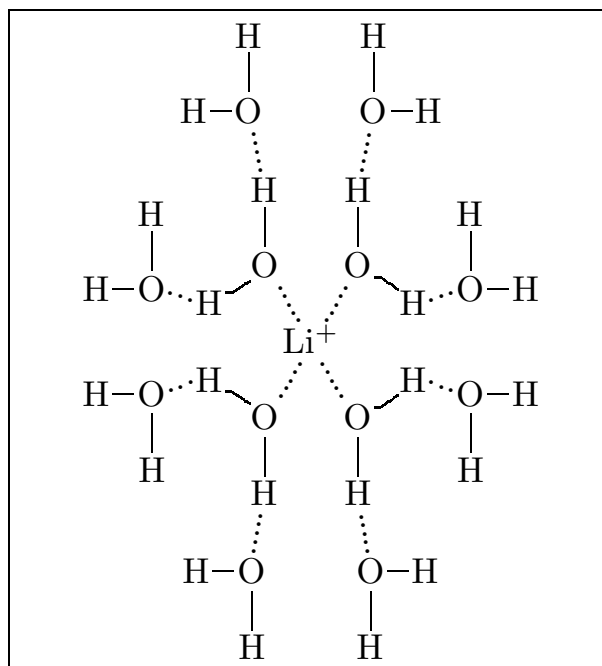


Figure 6: A lithium cation rendered by ChemStruct

Example 8: The ChemStruct Source for the Lithium Cation

```
\structure{\atom{~Li$^+$}
\side{\nwbelow\atom{O}
\side{\nsingle\atom{H}}nnwbelow\atom{O}
\side{\wsingle\atom{H}}
\nsingle\atom{H}}\swsingle\atom{H}
\wnwbelow\atom{O}
\side{\wsingle\atom{H}}\nsingle\atom{H}}
\side{\nebelow\atom{O}
\side{\nsingle\atom{H}}\nnebelow\atom{O}
\side{\esingle\atom{H}}
```

```

\nsingle\atom{H}\sesingle\atom{H}
\enebelow\atom{O}
\side{\esingle\atom{H}}\nsingle\atom{H}}
\side{\swbelow\atom{O}
\side{\ssingle\atom{H}}\sswbelow\atom{O}
\side{\wsingle\atom{H}}
\ssingle\atom{H}}\nwsingle\atom{H}
\swbelow\atom{O}
\side{\wsingle\atom{H}}\ssingle\atom{H}}
\side{\sebelow\atom{O}
\side{\ssingle\atom{H}}\ssebelow\atom{O}
\side{\esingle\atom{H}}
\ssingle\atom{H}}\nesingle\atom{H}
\esebelow\atom{O}
\side{\esingle\atom{H}}\ssingle\atom{H}}

```

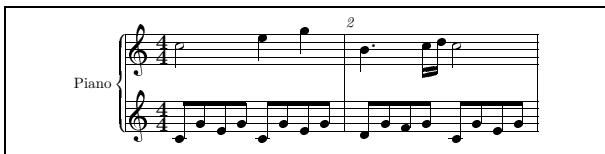


Figure 7: A little Mozart. . .

Example 9: The MusicT_EX Source for Figure ??

```

\begin{music}
\parindent lcm
\def\nbinstruments{1}\relax
\def\instrumenti{Piano}%
\nbporteesi=2\relax
\generalmeter{\meterfrac{4}{4}}\relax
\debutextrait
\normal
\tmps\Notes\ibu0f0\qh0{cge}
\tbu0\qh0g|\hl j\notes
\tmps\Notes\ibu0f0\qh0{cge}
\tbu0\qh0g|\ql l\sk\ql n\notes
\barre
\Notes\ibu0f0\qh0{dgf}}
\qlp i\notes
\notes\tbu0\qh0g|\ibb1j3
\qb1j\tb11\qbl1k\notes
\tmps\Notes\ibu0f0\qh0{cge}
\tbu0\qh0g|\hl j\notes
\finextrait
\end{music}

```

Another popular special-purpose application of T_EX is the production of transparencies, also called foils or slides. There are a few different options for this application.

3.3.1 SliT_EX

SliT_EX is part of the standard L^AT_EX distribution. Input to SliT_EX consists of a main file and a slides file. Individual slides are composed in a `slide` environment.

SliT_EX has provisions for black-and-white slides, color slides, and overlays. Unlike the other slide-making formats, which rely on `\special` printer commands¹⁴ to incorporate color, SliT_EX produces separate output pages for each color. For example, if you use red to highlight words on an otherwise black-and-white slide, SliT_EX will produce two output pages for the slide: one with all the

¹⁴The `\special` mechanism is a way of passing arbitrary information through T_EX to the DVI driver that will ultimately print the document.

¹⁵With the caveat that it still uses the Old Font Selection Scheme.

black text (excluding the words in red) and one with just the red words. Both of these pages will be printed in black. You must construct the colored slide by copying the pages onto colored transparencies and overlaying them. Producing slides with overlays in the same color is accomplished with a special ‘invisible’ color. This method of producing colored transparencies has been made obsolete by modern color printers. Of course, nothing prevents you from using the `\special` extensions of your DVI driver in SliT_EX to produce colored transparencies directly on a color printer.

SliT_EX has always been a separate macro package, distinct from and not 100% compatible with L^AT_EX. With the introduction of L^AT_EX 2_ε, SliT_EX is simply an extension package to the L^AT_EX 2_ε core format. Support for a separate SliT_EX format is being phased out.

3.3.2 FoilT_EX

FoilT_EX is an extension of L^AT_EX for producing slides. The primary advantage of FoilT_EX over SliT_EX is that it is completely compatible with L^AT_EX.¹⁵ Note, however, that the defaults in many cases are not precisely the same as L^AT_EX because of the radically different goal of FoilT_EX.

FoilT_EX provides support for running headers and footers, modified theorem environments for better mathematics in slides, support for AMS and PostScript fonts, and colors.

Color slides in FoilT_EX are handled by DVI driver `\special` commands (most commonly `dvips \specials`). However, FoilT_EX includes a number of new and extended features for better handling of color. See the section ‘T_EX in Color’ later in this chapter for a detailed discussion of using color in T_EX.

3.3.3 Seminar

The Seminar style is another alternative for producing slides and notes. Like FoilT_EX, the Seminar style is designed to work on top of L^AT_EX. (Seminar also works with AMSL^AT_EX.) The Seminar style is designed to produce output for a PostScript printer. It isn’t strictly necessary to produce PostScript output, but if you do not, many of Seminar’s features will be unavailable to you.

Seminar provides for a mixture of portrait and landscape styles and can support color using either a color-separation technique (similar to SliT_EX’s method) or direct use of PostScript color. In either case, the PSTricks macro package is required. (Consult the section ‘PSTricks’ in Chapter 6, *Pictures and Figures*, for more information.)

The Seminar style has support for a number of interesting options (including two-up printing of slides), automatic resizing by changing a magnification parameter, instructions for converting your SliT_EX slides, and several extensive demonstration files. Also included are explicit instructions for placing Encapsulated PostScript drawings in your slides.

3.4 T_EX in Color

With color printers and copiers becoming more common, the application of color, especially in transparencies, is more important than ever. Unfortunately, T_EX knows *nothing* about color.

A little reflection about the design of T_EX will make it clear why this is the case. T_EX produces device-independent output. Even when color printers are as common as ‘regular’ printers, if that ever becomes the case, it will always be true that color is an inherently device-dependent attribute. It does not make sense for T_EX to understand color. However, this does not prevent T_EX from using color.

At the lowest level, all that is required to use color in T_EX is some way of telling the printer ‘start printing in *<color>* here.’ This is easily accomplished with a `\special` command. In the discussion that follows, the *dvips* `\special` commands are used as concrete examples, but conceptually, any color printer can be used in this way.

3.4.1 Setting Up Color

Color support at the DVI driver level is provided by `\special` commands, but these are not typically convenient to enter directly into your document. Frequently, these commands are specified in terms of percentages of red, green, and blue (RGB color) or cyan, magenta, yellow, and black (CMYK color).

Higher-level support is provided by a collection of color control sequences. These sequences are loaded either by inputting the file *colordvi.tex* (in Plain T_EX, for example) or using the *colordvi* style file (in L^AT_EX).

dvips defines the colors in terms of ‘crayon names.’ If you need very precise control of the colors, you can adjust the precise mix of CMYK values in the file *colordvi.tex* after comparing the output of your printer with a standard scale (typically, the PANTONE scale). The following color names are standard in *dvips*.

Apricot	Emerald	OliveGreen	RubineRed
Aquamarine	ForestGreen	Orange	Salmon
Bittersweet	Fuchsia	OrangeRed	SeaGreen
Black	Goldenrod	Orchid	Sepia
Blue	Gray	Peach	SkyBlue
BlueGreen	Green	Periwinkle	SpringGreen
BlueViolet	GreenYellow	PineGreen	Tan
BrickRed	JungleGreen	Plum	TealBlue
Brown	Lavender	ProcessBlue	Thistle
BurntOrange	LimeGreen	Purple	Turquoise
CadetBlue	Magenta	RawSienna	Violet
CarnationPink	Mahogany	Red	VioletRed
Cerulean	Maroon	RedOrange	White
CornflowerBlue	Melon	RedViolet	WildStrawberry
Cyan	MidnightBlue	Rhodamine	Yellow
Dandelion	Mulberry	RoyalBlue	YellowGreen
DarkOrchid	NavyBlue	RoyalPurple	YellowOrange

3.4.2 Using Color

After *dvips* has loaded *colordvi*, typesetting text in color is simply a matter of using the appropriate color control

sequence. For example, to typeset something in red, use the `\Red` control sequence in your document, like this:

```
\Red{something in red}
```

Alternatively, you can change the default text color with the `\textcolor` control sequences. To make default color for all text blue, enter:

```
\textBlue
```

To change the background color, use the `\background` macro. For example, to make the current and all future pages yellow, enter:

```
\background{Yellow}
```

You can enter a precise color by specifying it in terms of its CMYK components. The `\Color` and `\textColor` macros exist for this purpose. To typeset some text in a color that is 25% cyan, 35% magenta, 40% yellow, and 10% black,¹⁶ enter:

```
\Color{.25 .35 .4 .1}{some text}
```

3.4.3 Now I’ve Got Color, but I Need Black and White!

If you have reason to print a colored T_EX document on a black and white printer, you don’t have to tear out all of the color commands. *dvips* includes a *blackdvi* file (analogous to *colordvi*—an input file or style file depending on your macro package), which translates all color commands into black and white.

Alternatively, ‘good’ implementations of PostScript in a black and white printer should translate all colors into shades of grey. This can be an inexpensive way to preview a color document. Most screen previewers simply ignore color commands so they print in black and white even if the document is colored.

3.4.4 Color Under L^AT_EX 2_ε

At the time of this writing, the L^AT_EX 2_ε team has not officially adopted a standard for using color. However, it is likely to follow a slightly different model than the one described above. The final design should provide color selection commands that are device-independent at the DVI driver level (in other words, the color commands will not insert device-specific commands, like snippets of PostScript, into the DVI files).

¹⁶I made these numbers up. I take no responsibility for the artistic merits (or lack thereof) of the resulting color.

3.4.5 Color Is Subtle

Color commands implemented as `\special` commands may introduce occasional problems. For example, if T_EX introduces a page break in a paragraph that you have typeset in yellow (`\Yellow{This is a long paragraph...}`), the resulting output may print the page footer (and even the header) in yellow, although that was not intended.

Circumventing these problems may require careful use of color commands in front of text that you want to appear black. For example, in Plain T_EX the difficulty described above can be avoided by specifying that the page number should be printed this way:

```
\footline{\Black{\hss\tenrm\folio\hss}}
```

This definition guarantees that the page number will be set in black, and because it is a local color change, colored text can flow across the page around it.

You may want to make sure that other typographic elements are printed in the current global color (which may vary). *dvips* provides a local color macro called `\globalColor` for that purpose. Every time the text color is changed globally (with a `\textColor` command), `\globalColor` is redefined to print text in that color.

3.4.6 Further Reading

Read the documentation for your DVI driver carefully with respect to color. Because it is device-dependent, there is a lot of room for interpretation, and it may not always be obvious why some things appear the way they do. And DVI drivers are free to implement color in any way they choose.

Review of 'Making T_EX Work'

Paul Thompson

Department of Psychiatry
Case Western Reserve University
Cleveland, OH 44106, USA
pat@po.CWRU.Edu

Title : Making T_EX Work
Author : Norman Walsh
Publisher : O'Reilly and Associates
Date : April, 1994
ISBN : 1-56592-051-1
Pages : 518
Price : US \$29.95

-- * --

I have recently received a review copy of Making T_EX Work, by Norman Walsh (published by O'Reilly Associates). From my perspective, I am a somewhat to rather sophisticated user of L^AT_EX/T_EX.

Making T_EX Work (MTW) is another book along the lines of The L^AT_EX Companion (TLC). MTW takes a wider view of the entire T_EX/L^AT_EX enterprise than does TLC, in that MTW involves editing, plotting, and all the aspects of using T_EX, rather than TLC, which involves merely L^AT_EX. The chapter titles for MTW are The Big Picture, Editing, Running T_EX, Macro Packages, Fonts, Pictures and Figures, International Considerations, Printing, Previewing, Online Documentation, Introducing METAFONT, Bibliographies, (Tools) Non-Commercial Environments, Commercial Environments, T_EX on the Mac.

Generally, I like the book and have found it to be useful. From the user standpoint, it is probably most useful for the somewhat naive, but somewhat experienced user. For instance, consider the chapter on Editors. Subheaders are What can an editor do?, GNU Emacs, AUC-T_EX, Multi-Edit, Brief, MICROEmacs, epm., Other Tools, T_EX as a compiler, Spellchecking, Revision Control, T_EX Shells. The first sections discuss various capabilities of editors. I found the section on GNU Emacs and AUC-T_EX to be most helpful, not being familiar with AUC-T_EX prior to

reading the book (although I was an emacs user for a long while). The other editors are discussed as well at the same level - very interesting for the person not familiar with the product, not enough detail for anyone else.

Similarly with the Fonts chapter. The matter of fonts is exceedingly poorly described in T_EX/L^AT_EX as a whole, IMVHO. I did find this chapter describing fonts in a clear and simple fashion. The chapter begins with a discussion of the process by which T_EX locates font information. Some examples of font information are presented. A variety of issues are considered, including magnification, font size, location of font information, and other issues of standard font information. Some more advanced issues are also discussed - NFSS2, Postscript fonts under NFSS, encoding vectors, virtual fonts and automatic font generation. My overall impression is that this information is very helpful to persons like myself who are still a little confused when fonts are discussed. The term "Cork encoding" is defined in the book, although not in this chapter. So, I generally found the information here quite helpful.

I must wrap this up at this point. My overall impression of the book is pretty positive. The author has made a real effort to tie in a lot of disparate pieces of information. This information is all out there, but very hard to pull together. However, the book is not 100% problem is that the book lacks a critical overview, but this is probably not the author's interest. For instance, in the discussion of T_EX shells, the author mentions 5, but doesn't spend any time comparing and contrasting. So, if you were trying to select a T_EX shell, you couldn't just use this book. I guess that I am saying that the book is not the Consumer's Reports of T_EX stuff, but is that a problem? I dunno. I did find book very helpful at times.

Note: I have no financial interests in O'Reilly and Assoc. I did get the review copy free.

Typesetting Greek texts by \TeX and \LaTeX

— a summary of all tools available —

Dimitrios Filippou*

Jan 28, 1994

Abstract

This paper contains some information on whatever exists for typesetting by \TeX documents which contain entirely or in part Greek text. Those who would like to typeset Greek texts not by plain \TeX , but rather with \LaTeX , may find this short article quite useful as well.

This article is not a cookbook for Greek \TeX ing. In fact, all that is presented here, is just a summary of starting pointers for someone who wishes to typeset the poems of Sappho (late 7th-early 6th c. B.C.) or Odysseus Elytis (1911–) in their original script. My personal experience during the past few years, says that even these pointers can be extremely helpful when you are striving to typeset real Greek texts in an environment where Greek characters are ‘something to be used only in mathematical formulae’.

Therefore, the following text contains some information on

- Greek fonts created for \TeX by METAFONT (this will be of great interest to the few ones who like to play with METAFONT),
- complete \TeX/\LaTeX packages for typesetting Greek documents (the quick solution for anyone who wants to typeset Greek texts with \TeX or \LaTeX),
- where should one seek help in case he has problems to typeset Greek with \TeX or \LaTeX (the Greek connections :-)), and
- the future of typesetting Greek documents by \TeX (just a short note on where \TeX ing Greek texts is heading to).

This short article is based on a note that I posted for the first time in the USENET newsgroup ‘comp.text.tex’ in February 1993. That posting has been apparently archived under the name ‘help/greek.faq’ at all CTAN sites:

```
ftp.shsu.edu (192.92.115.10),
ftp.dante.de (129.206.100.192), and
ftp.tex.ac.uk (134.151.79.32).
```

I hope that the gentle administrators of the network of \TeX archives will replace that file by the present, more complete article.

1 Public domain Greek fonts for \TeX

The first Greek fonts of \TeX were those that D.E. Knuth created himself and which have become integral part of

every \TeX distribution. Of course, these fonts have been designed to be used in mathematical expressions, but in the absence of any other alternative, they may be used to typeset a few words of regular Greek text. The result though will be far from perfect, especially if the text is in ancient Greek, full of accents and diacritical marks. In such a case, you must look for some other Greek fonts.

Silvio Levy (Princeton University, New Jersey, USA) was the one who created the first family of fonts for typesetting Greek text with \TeX . Levy’s fonts were created by METAFONT and appeared by the mid-80s, just as \TeX version 3.0 started coming out. That was an 8-bit font family and included regular, slanted, bold and typewriter typefaces. Levy made his Greek fonts after the Didot design, a typeface which was originally created two centuries ago by the famous Didot printers in Paris and which is still used extensively by Greek printers today. Among Greek printers, this Didot typeface is known as ‘*apla*’ (plain), but often outsiders call it incorrectly ‘Greek roman’.

The METAFONT source code of Levy’s fonts is still available in some FTP sites (usually in font collections). Nonetheless, it is not worth to bother with these METAFONT sources, because Levy’s fonts have been surpassed by the similar ones that were subsequently created by Yannis Haralambous (Villeneuve d’Ascq, France).

Yannis Haralambous’ family of Greek fonts looks the same as Levy’s one, i.e., it is of the Didot kind, but uses a different 8-bit coding scheme. It contains five typefaces: regular, slanted, bold, ‘*italics*’ (these ‘*italics*’ are kind of pseudo-italics based on the Greek math characters of the Computer Modern fonts) and small caps. Within the small caps of Haralambous, you will find some extra characters such as digamma, qoppa and sampi, which are useful in writing Greek numerals. The METAFONT sources are available by FTP from many ‘*fonts/greek*’ collections and some old ‘*babel*’ collections such as:

*Dimitrios Filippou wrote this short note while he was a doctoral student at the Department of Mining and Metallurgical Engineering of McGill University, Montréal, Québec, Canada. His permanent address is: Kato Gatzia, GR-385 00 Volos, Greece.

```
ymir.claremont.edu (134.173.4.23),
directory: tex.babel.greek.yannis.
```

Together with his fonts, Yannis Haralambous has also put a few essential macros to allow the easy use of the fonts.

In the time between Levy's and Haralambous' works, Brian Hamilton Kelly (Royal Military College of Science, Swindon, UK) also presented a family of Greek fonts. Apparently these fonts (Greek 'roman', 'italic', bold and typewriter) were created out of the Greek characters of the Computer Modern math fonts. I have never tested Hamilton Kelly's fonts, but, as their author says, they were created only for modern uni-accent Greek; they will not work for multi-accent ancient or modern Greek. The METAFONT sources of these fonts are available by FTP from:

```
ymir.claremont.edu (134.173.4.23),
directory: tex.babel.greek.hamiltonkelly.
```

The two public domain Greek \TeX packages which have been put together by Moschovakis and Dryllerakis and which are discussed later in this article, are heavily based on the original work of Levy and Haralambous. In terms of fonts, both these public domain Greek \TeX packages include Levy's and/or Haralambous' original work only slightly modified. Moschovakis, for example, has added his own experimental 'Greek italics' and 'Greek sans serif'.

2 Commercial Greek fonts for \TeX

SCHOLAR \TeX is a commercial \TeX package that has been created by Yannis Haralambous. It includes fonts and macros for modern Greek, ancient classical Greek, ancient epigraphical Greek and ('in the near future', according to the author) Byzantine Greek. Indeed, the scholar may find this package extremely useful as, in addition to Greek, it includes fonts for a number of other non-Latin alphabets: Arabic, old Turkish, Persian, Urdu, Malay, Armenian, etc. SCHOLAR \TeX is also the only package available at this moment that contains complete hyphenations tables for ancient classical Greek and modern Greek (hyphenation rules are not the same for ancient and modern Greek). The price of SCHOLAR \TeX is US \$200 for individuals (US \$100 additional for METAFONT sources) and US \$500 for institutions (includes METAFONT sources). For orders or more information, you can contact:

Yannis Haralambous
101/11 rue Breughel,
59650 Villeneuve d'Ascq,
France
e-mail: yannis@gat.univ-lille1.fr
fax: +33 20.91.05.64

You must specify to the author of SCHOLAR \TeX what is the operating system where you intend to use his product.

C. Mylonas and R. Whitney have also created Greek fonts for \TeX . Their fonts designed after the Times-Elsevier Greek typefaces, which are very different from the Didot one. The most striking thing of Mylonas and Whitney's work, which has been published in the TUGboat [vol. 13

(1992), no. 1, pp. 39–49], is that it contains the most complete Greek character set I have ever seen: 360 characters in total! The creators of these last fonts have reported that they are continuously improving their work and that they have already prepared macros and hyphenation tables for modern Greek and ancient Greek. Nonetheless, they have decided not to distribute or sell their fonts for the moment.

3 Public domain Greek \TeX packages

At this moment, there exist two complete public domain packages with fonts and macros for typesetting Greek documents by \TeX or \LaTeX . These are:

1. 'greetex' version 2.0 (?) by Yiannis Moschovakis (UCLA, Los Angeles, USA), and
2. 'GreekTeX' version 3.1 by Kostis Dryllerakis (Imperial College, London, UK).

The names of these packages are indeed identical. Hence, in order to distinguish the one from the other, I have denoted the first one with lowercase letters.

Both these packages include a number of 8-bit fonts: regular, slanted, typewriter, etc. They also include macros for typesetting ancient or modern, uni-accent or multi-accent Greek by either plain \TeX or \LaTeX . Greek hyphenation tables are also provided in these packages. However, these tables have been prepared according to the hyphenation rules of modern Greek; therefore, they may give a few erroneous results with ancient Greek texts.

Particularly for \LaTeX users, Moschovakis has included in his 'greetex', a style file for entirely Greek articles. Dryllerakis, from his side, has added a Greek article and a Greek book style file in GreekTeX. These Greek style files are in reality the standard \LaTeX article and book style files with just few modifications, mostly in titles. GreekTeX also includes a 'greek.sty' which comes very handy to those who want to include only a few Greek quotes in their documents.

The Greek \TeX packages are accompanied by 'filters' for character translation. With these filters, the packages can be used under any system which accepts standard 7-bit ASCII encoded input such as UNIX, DOS, etc. However, each of these packages was originally created under a different operating system. Consequently, 'greetex' is more friendly to use on a DOS machine, because it allows the user to type in Greek or Latin directly. On the other hand, I find GreekTeX a better choice for UNIX running machines, since at many UNIX terminals you cannot edit a document with Greek characters encoded at ASCII positions above 127. For DOS machines, GreekTeX works very well with the 'texconv' filter of emTeX. Kostis Dryllerakis has also reported that his GreekTeX has been used successfully on a Macintosh (I would think with OzTeX).

The fonts of 'greetex' include the basic Computer Modern Latin characters at their original positions (ASCII below 127). Contrary to this, the fonts of GreekTeX do not include any Latin characters, but only Greek ones. For that reason, \TeX users who want to avoid duplicating the font

files in their disks, should go for Dryllerakis' package. I would recommend Moschovakis' 'greetex' to those who want to typeset entirely (or or almost entirely) Greek texts by \TeX on their DOS machine. However, those who will pick up Moschovakis' package should be aware that their '.tex' file may not be easily transferred from one machine to another. Portability seems to be the biggest advantage of the other package, Dryllerakis' Greek \TeX , even if editing something for Greek output by Greek \TeX is a bit more tedious. Greek \TeX also seems to be the more suitable Greek \TeX package for the New Font Selection Scheme (NFSS) and the upcoming \LaTeX version 3.0.

For those who are interested, Moschovakis' 'greetex' is available by FTP from:

```
math.ucla.edu (128.97.4.254)
                                [original site],
directory: pub/moschovakis/greetex
```

and in all CTAN sites (these sites are listed at the beginning of this article) in the directory:

```
fonts/greek/moschovakis.
```

Dryllerakis' Greek \TeX is available from:

```
laotzu.doc.ic.ac.uk (146.169.2.9)
                                [original site],
directory: public/tex
```

and in all CTAN sites in the directory:

```
fonts/greek/kd.
```

It is worth to add at this point that each Greek \TeX package takes about 2 Mb of space on the hard disk of my PC. This includes documentation files, files with macros, format files, '.tfm' font files and '.pk' font files for a HP Laserjet printer.

In addition to these packages, a few fellow Greeks have mentioned to me that at two Universities in Greece, the University of Patras and the University of Crete, there have been created, installed and distributed to students some other Greek \TeX packages. From fragments of information I gathered, I have come to the conclusion that these packages have been built up on Levy's original work. Unfortunately, these 'very Greek' \TeX packages have not been archived at any FTP site and I could not find a copy of them.

4 Contacting the Greek \TeX community

Most of the fonts, macros, and packages that were described in the previous paragraphs have been created by people who have showed much enthusiasm on typesetting Greek by \TeX , but without any intention of making money out of this story. Their work is available to the public at no-charge, but it can not be guaranteed that it is error-free or that everything will be OK under any \TeX ing circumstances. Sometimes, you may have to do a bit of hacking and, if like me you are not a TeXnician, you may find yourself often frustrated.

Thus, in the case you face problems with typesetting Greek with \TeX , but also if you have new ideas about this subject, you should join the ELLHNIKA mailing list. To do this,

just send a message to `LISTSERV@DHDURZ1.BITNET` by e-mail. This message must contain only one line in its body:

```
SUBSCRIBE ELLHNIKA <your 1st name> <your last name>
```

The list has currently about 80 subscribers with Yannis Haralambous being the list owner. Once you are a subscriber to this list, you may send your questions, your problems, or your ideas to `ELLHNIKA@DHDURZ1.BITNET`. Ultimately, you may also post your problem or request to the relevant newsgroup of USENET, namely 'comp.text.tex'.

5 The future of Greek \TeX

The fact that there exist two Greek \TeX packages now available in public domain reveals a subtle fight among Greek \TeX ncians for the future of Greek \TeX . On the one side, there are those who want pure Greek fonts (i.e., not attached to any Latin character set) harmonised with the NFSS and the upcoming \LaTeX version 3.0. This is what can be achieved with a package like Dryllerakis' Greek \TeX . The problem with the pure Greek fonts is that the basic Greek characters are encoded in 7-bit ASCII positions: you have to type q to get a 'chi', y to get a 'psi', etc. Editing the \TeX / \LaTeX source code becomes rather difficult and unfriendly, especially for a new user of Greek \TeX who may be accustomed to a certain Greek keyboard layout. The use of filters may make life slightly easier, but it is not the same as typing directly Greek into your \TeX / \LaTeX code.

This is exactly the argument of the 'other side', those who prefer more machine-specific packages like Moschovakis' 'greetex', which allow the user to edit a \TeX / \LaTeX source code directly in Greek on his/her machine. But here, the biggest problem is portability, because almost every computer maker uses his/her own Greek upper ASCII coding scheme.

In any case, time has come for some standardisation at least on Greek \TeX character encoding. The \TeX Users Group (TUG) realised this necessity not only for Greek \TeX , but also for using \TeX with any language other than English. TUG therefore formed the Technical Working Group on Multilingual Coordination (TWGMLC) for the development and standardisation of a truly multilingual \TeX / \LaTeX . Yannis Haralambous took the responsibility of leading the TWGMLC and Silvio Levy has become the chair of the Greek subgroup.

More specifically, the target of the TWGMLC is to provide every user of \TeX or \LaTeX with a minimum ' \TeX Language Package' which will include: (i) standardised software (fonts and macros), and (ii) a minimum documentation about \TeX and \LaTeX in the language of the user. For the time being, the TWGMLC has not released anything from its project. Soon though, and hopefully at the time of the release of \LaTeX version 3.0, we may have the first manuals of Greek \TeX or \LaTeX in Greek!

Beyond the effort of the TWGMLC, it is worth noting that Brian Hamilton Kelly, and C. Mylonas and R. Whitney have also reported considerable improvements on their own fonts and macros, but they are not releasing them yet. Yannis Haralambous is working on Byzantine Greek, a new true Greek typewriter and a new sans serif font. And Kostis Dryllerakis has promised that the next version of his GreekTeX will be NFSS-compatible and that his fonts will include the almost obsolete Greek characters digamma, qoppa and sampi. 'Idomen' (We'll see. . .).

Acknowledgements

If I were to thank people who helped me to put this information together, I would have to make a long list of names and e-mail addresses. Everyone who showed me even tiny details about \TeX and \LaTeX has contributed in his way into putting this information together. However, there are four people who have helped me considerably and whom I feel obliged to name:

Leonidas J. Irakliotis (l.irakliotis@ieee.org),
Yiannis N. Moschovakis (ynm@math.ucla.edu),
Kostis Dryllerakis (kd@doc.ic.ac.uk), and
Yannis Haralambous (yannis@gat.univ-lille1.fr).

Springer authors do it with T_EX

Angelika Binding

Springer-Verlag Heidelberg, Germany,
binding@vax.ntp.springer.de

April 1994

At Springer-Verlag the first experiments with T_EX began in 1988, at the instigation of the editors in the physics' department. Our production staff were initially sceptical, feeling that Springer's high-quality books had to be typeset in Monotype Times and that authors would never achieve the required standard of presentation. Today, however, some of the most severe critics are enthusiastic converts to T_EX.

1 General concept

T_EX is used to typeset both books and journals articles in mathematics, physics, engineering and, of course, computer science.

Two strategies are possible: the author delivers either camera-ready pages with line-art integrated or pasted in, or files that are T_EXed and the result output on a phototype-setting machine.

The advantages of using authors' files for publishing have been stated many times: There is less proofreading, as there is no rekeying which may introduce new mistakes. Authors also do not have the problem of how to make clear to the typesetter exactly which symbol they want to be used in a certain context and how a specific formula should be arranged, because they have complete control over these aspects. Publication times are reduced, which is especially valuable in journals. For the reader the most striking advantage is the great improvement in typesetting quality for camera-ready books. These used to be reproductions of typewritten pages with mathematical symbols often drawn in by hand. Now their appearance more and more resembles that of conventionally typeset books. With the advent of 600 dpi laser printers the quality gap between the two methods has narrowed considerably.

However, T_EX files do not automatically end up as beautifully made up pages. Although any T_EX user can produce readable documents with sometimes quite complex formulas, it takes someone with a good knowledge of page layout to exploit T_EX to full advantage. It was this misunderstanding, namely that anybody could make beautiful books with T_EX, that initially has brought T_EX into discredit.

To assist their authors, publishers usually produce small brochures that explain their basic typographical rules and conventions, but it takes quite an effort to implement those rules into T_EX code. Sometimes authors take great pleasure

in writing macros to produce the layout for their project, but they should not be obliged to first take a course in typography before starting to type their manuscripts.

Instead, we provide them with macro packages that enforce the layout that is needed for a certain book series or journal. Here we always have the trade-off between

- maintaining typographical standards
- giving authors the flexibility they need for their work
- minimizing development costs

Authors should not have to bother with how much white space is necessary before and after a theorem or how to avoid a section heading at the bottom of a page, but they need the choice of different referencing systems and numbering schemes. In addition, typographical conventions vary from one discipline to another. For a journal article the typographical guidelines are of course stricter, to ensure a uniform appearance of the journal than they are for a monograph.

We provide macro packages for L^AT_EX as well as plain T_EX, both for journal articles and for books. Nowadays L^AT_EX is preferred by most authors, even though intensive use of the automatic referencing facilities offered by L^AT_EX often leads to serious memory shortage. Tables 1 and 2 list all macro packages currently available.

All macro packages come with detailed documentation that covers installation and use of the macros and features valuable tips on how to improve the appearance of the document. Sample files are often included, along with a reference card for the impatient. We used to distribute the AMS fonts with our packages but lately we feel this is necessary only in exceptional cases.

2 How to obtain the macro packages

All macro packages are available (free of charge) to all Springer authors. Originally we handed out diskettes, but now all macro packages are on a mailserver and on an ftp/gopher server for fast and easy distribution.

2.1 Mailserver

The mailserver has the e-mail address

svserv@vax.ntp.springer.de

All e-mail requests to this server must be in a specific format as the requests are handled by a program rather

Name	Vers.	supported fonts ^a	page size (cm)	to be used for
CPMoMu01	1.0	cm	11,7 x 18,9	monographs and proceedings
CP-EMS	1.1	cm	11,8 x 19	<i>Encyclopedia of Mathematical Science</i> book series
PJour1	1.0	cm/ps	12,2 x 19,5	articles for the journals - <i>Acta Informatica</i> - <i>Archive for Mathematical Logic</i> - <i>Communications in Mathematical Physics</i> - <i>Economic Theory</i> - <i>Inventiones Mathematicae</i> - <i>Journal of Mathematical Biology</i> - <i>Mathematische Annalen</i> - <i>Mathematische Zeitschrift</i> - <i>Numerische Mathematik</i> - <i>Probability Theory and Related Fields</i> - <i>Theoretica Chimica Acta</i>
PJour1g	1.0	cm/ps	12,2 x 19,5	articles for arbitrary Springer journals with one-column layout e.g. - <i>Calculus of Variation</i>
PJour2	1.0	cm/ps	17,8(8,6) x 24	articles for journals with two-column layout e.g. - <i>Informatik Forschung und Entwicklung</i> - <i>OR Spektrum</i>
PJour2g	1.0	cm/ps	17,8(8,6) x 24	articles for journals with two-column layout e.g. - <i>Multimedia Systems</i> - <i>Machine Visions & Applications</i>
PLNCS	1.1	cm	12,2 x 19,3	contributions to volumes of <i>Lecture Notes in Computer Science</i>

^a cm = Computer Modern fonts, am = Almost Computer Modern fonts, ps = PostScript Times fonts

Table 1: Macro packages for plain T_EX

Name	Vers.	supported fonts	page size (cm)	to be used for
CLMoMu01	1.0	cm	11,7 x 18,9	monographs and proceedings
LJour1	1.0	cm/ps	12,2 x 19,5	articles for journals with one-column layout see PJour1 and PJour1g
LJour2	1.0	cm/ps	17,8(8,6) x 24	articles for journals with two-column layout, see PJour2 and PJour2g
LLNCS	1.1	cm	12,2 x 19,3	contributions to volumes of <i>Lecture Notes in Computer Science</i>

Table 2: Macro packages for L^AT_EX

than a person. An e-mail sent to order a macro package has to contain a line of the form

```
get directory\filename.zip
```

e.g.

```
get \tex\latex\clmomu01.zip
```

To ensure safe transmission over the nets, the files are uuencoded and pkzipped. Thus the filenames end with .zip.

2.2 Gopher and ftp

For users with access to ftp and gopher, these transmission methods are, of course, much more convenient. Our server has the name *trick.ntp.springer.de*. For ftp the user

is ID *anonymous*. Please enter your e-mail address at the password prompt.

3 Procedure for books

Authors who wish to use T_EX when preparing a book for publication with Springer should contact the editor as early as possible. Once it has been decided whether the book should be produced camera-ready by the author or typeset from files supplied by the author, he receives a suitable macro package. Authors who prefer to use their own macros are usually free to do so. In either case we request that the author send in a few sample pages so that we can check that all requirements are met.

4 Procedure for journals

One of the first Springer journals to use T_EX was *Semigroup Forum*, where the editor himself wrote a set of macros. The appearance of this journal which is produced camera-ready, has thus improved greatly at no additional cost.

T_EX is also used for about 30 typeset journals. As many of these are typeset with Times fonts, our article macros offer the option of using them together with PostScript Times fonts should the latter be available. This ensures that the page proofs produced by the author have nearly the same line and page breaks as the final printed version, which is very important when it comes to typesetting complex displayed formulas, where only the author can decide where to insert line breaks.

After the article has been accepted for publication the author is requested to send the final version of the T_EX file together with a hard copy and all figures that should be included to the production editor at Springer-Verlag. Here copy editing changes are incorporated and often improvements in page make-up have to be carried out. A PostScript file is then sent to the typesetting company for exposure on film.

5 A look into the future

To further improve the service to our authors, we accept authors files over Internet, as some authors find e-mail more convenient than converting files to a DOS diskette.

We always request a hard copy of the article to be able to check whether the delivered file actually contains the final version and that it is intact. Together with this print out, we receive the originals of the figures that the author wishes to include in his publication.

Files transmitted via e-mail are sometimes corrupted. We therefore ask that they be uuencoded. Not all authors are familiar with this program. With ftp, however, corruption of files is not a problem.

Lately more and more authors supply us with EPS files of their figures which we then incorporate in the T_EX file. This enables us to build up a fully electronic archive of articles which can be used to publish the journals in electronic form as well.

The first journals, which is entirely done in T_EX and which we also publish as an electronic version is *Numerische Mathematik*, a well known journal on numerical analysis. Subscribers to the paper version may order the files of the articles as well. Initially, delivery as T_EX resp. L^AT_EX files with integrated image files (EPS or TIFF) is planned, but other formats are possible as well. We are currently conducting a market research on this subject. Persons and institutes interested in testing this offer are requested to contact our Electronic Media Department (e-mail:em-helpdesk@springer.de).

T_EX-verwerking bij PRAGMA

J. Hagen

Onderwijskundig Bureau voor Advies- en Ontwikkelwerk,
Postbus 125,
8000 AC Zwolle

11 januari 1994

Waarom T_EX

Bij PRAGMA kunnen bedrijven en instellingen terecht met vragen en problemen op het gebied van onderwijs en opleidingen. De werkzaamheden van PRAGMA bestaan onder andere uit: het ontwikkelen van opleidingen en opleidingsmateriaal, het ontwikkelen van beleid met betrekking tot opleiden en het uitvoeren van gericht onderzoek. Daarnaast houdt PRAGMA zich in toenemende mate bezig met het ondersteunen van documentatie op de werkplek.

Omdat de papieren produkten vaak de enig tastbare produkten zijn besteden we veel zorg aan de vormgeving. Daarbij worden als uitgangspunt de eerder genoemde aspecten als onderhoudbaarheid, onafhankelijkheid en controleerbaarheid gehanteerd.

De eerste jaren heeft PRAGMA haar materialen hoofdzakelijk in een t_ypletter vormgegeven. Kijkend naar de betaalbare mogelijkheden die op dat moment voorhanden waren, werd ons motto: beter een goede t_ypletter dan slecht zetwerk. Geïnspireerd door het T_EXbook zijn eind jaren '80 T_EX en L^AT_EX aangeschaft. Hoewel het op Amerikaanse leest geschoeide uiterlijk van de in L^AT_EX aangemaakte teksten ons niet aansprak, werd toch steeds meer gebruik gemaakt van dit systeem. In de loop van de tijd ontstond er langzaam een eerste schil rond L^AT_EX. Onbekend met het bestaan van gebruikersgroepen, bleek het aanpassen van style-files voor een beginneling niet eenvoudig. (Overigens is L^AT_EX zelf een uitstekend gedocumenteerd pakket).

Niet tevreden met L^AT_EX werd, na een korte flirt met achterevolgens L^AM_S-T_EX, macroT_EX en inrsT_EX, besloten een eigen systeem te ontwikkelen bovenop T_EX. Alleen table en picT_EX zouden worden gebruikt (ook dit zijn uitstekend gedocumenteerde pakketten). Terugkijkend kunnen als belangrijkste redenen voor deze in 1991 genomen stap worden genoemd:

1. de wens om meer te kunnen dan op dat moment (standaard) kon, zoals bijvoorbeeld het in de tekst opnemen van in tekenpakketten aangemaakte figuren
2. de wens om meer vrijheidsgraden in vormgeving, maar dan wel met behulp van een uitbreidbaar, inzichtelijk en consistent systeem
3. de wens om in geval van specifieke gevallen terug te kunnen vallen op T_EX

4. de behoefte aan een vergaande integratie en samenwerking tussen in huis ontwikkelde programmatuur en T_EX
5. de noodzaak om relatief onervaren computergebruikers (werknemers en stagiaires) snel goed vormgegeven teksten te laten produceren
6. de daaruit volgende eis van nederlandsstalig zijn
7. het besef dat er meer in de wereld is dan wiskundig zetwerk
8. het gegeven dat het grote aantal hulpfiles dat door sommige pakketten worden gegenereerd het overzicht niet bevordert
9. de wens documenten te kunnen opdelen en groeperen op een wijze die projectmatig beheer van teksten mogelijk maakt

Deze lijst is niet volledig, maar geeft wellicht een redelijk beeld. De door verschillende gebruikers op netwerken aangeboden macro's zijn voor ons helaas zelden bruikbaar, omdat deze meestal bovenop L^AT_EX geschreven zijn. Er was dan ook nauwelijks teruggevallen op reeds beschikbare macro's. Een uitzondering hierop vormen enkele in gebruikersbladen gepubliceerde routines voor het plaatsen van figuren.

De werkomgeving

Onze werkomgeving heet, hoe kan het ook anders, T_EXedit. Met de binnen dit systeem gebruikte, in huis ontwikkelde tekstverwerker, werken we nu zo'n zes jaar. Destijds is zij ontwikkeld omdat er weinig programma's voorhanden waren die snel, betrouwbaar, consistent en ASCII waren. Binnen T_EXedit kunnen T_EX-files worden aangemaakt, gecontroleerd en, al dan niet gedeeltelijk, verwerkt. Door middel van kleur kan de gebruiker T_EX-commando's en argumenten visueel controleren. Daarnaast kunnen controles op symmetrisch gebruik van haakjes en commando's worden uitgevoerd.

T_EXedit ondersteunt een logische opbouw van teksten in projecten, omgevingen, produkten en onderdelen. Het programma maakt daarbij gebruik van enkele structurerende T_EX-commando's (zie figuur 1).

<p>Tekst : hoofd04</p> <pre> \startonderdeel hoofd04.tex \projekt stroom.tex \produkt cursist.tex \hoofdstuk{Wrijvingsloze stroming} \startintro De stroming van gas wordt bepaald door een aantal theoretische hoofdwetten. In dit hoofdstuk worden de volgende drie wetten behandeld: \startopsomming[opeelkaar] \som de wet van behoud van massa \som de wet van behoud van energie \som de algemene gaswet \stopopsomming Hoewel in paragraaf~\in[toestand] al werd gesteld dat er altijd wrijving optreedt, verwaarlozen we dat verschijnsel </pre>	<p>Wijzig</p> <pre> hoofd04 prj stroom prd cursist </pre> <p>Opties</p> <pre> Shift F6/F12 Ctrl F6/F12 Alt F6/F12 </pre>
--	--

Het is mogelijk een tekst snel visueel op spelfouten en storend taalgebruik te controleren. Foute woorden krijgen een rode kleur en zijn daardoor snel te vinden. Woorden van minder dan vier letters worden niet gecontroleerd, omdat bijna alles goed is (is, in en ik zijn stuk voor stuk goed geschreven woorden). Omdat goede woorden bij het intypen van een woord groen worden, vallen de korte woorden goed op. Met deze vorm van (real-time) controle, zijn teksten relatief snel arm aan spelfouten te maken, mede omdat ingetypte tekst direct wordt gecontroleerd. Daarbij worden accenten zonodig automatisch geplaatst. Overigens krijgen speciale groepen woorden ook een kleur, zodat snel een beeld van de schrijfstijl kan worden gekregen: actief of passief, neutraal of persoongericht.

Andere kenmerken van T_EXedit zijn: het door T_EX halen van (geselecteerde delen van) een tekst, het leveren van helpinformatie over commando's, het geven van overzichten van figuren en het tussenvoegen van (een of meer) voorgedefinieerde commando's (templates).

In tegenstelling tot de meeste systemen, gebruiken wij slechts één file per tekst voor overzichten, gesorteerde lijsten, indices, referenties enz. Het programma T_EXutil draagt zorg voor het sorteren van een en ander. Er kunnen meerdere lijsten en indices worden aangemaakt.

T_EXutil ondersteunt het selectief verwerken van delen van de tekst. Het is bijvoorbeeld mogelijk formules, vragen en/of antwoorden onafhankelijk van de plaats in de ASCII-tekst op te roepen: op dezelfde plaats, aan het eind van een hoofdstuk, in een apart deel enz. Het programma kan ook een overzicht van figuren (eps) genereren. De gegevens in deze file worden gebruikt bij het plaatsen en het op verschillende manieren kunnen schalen van de figuren.

Naast T_EXedit en T_EXutil gebruiken wij nog enkele programma's. Met T_EXadres kan een adresbestand worden

onderhouden met behulp waarvan brieven naar personen of (werk)groepen kunnen worden verstuurd. Met T_EXform kunnen argumenten van T_EX-commando's interactief worden 'ingevuld' en 'geT_EXed'. Overzichten van figuren, adreslijsten, labels voor mappen enz. zijn hierdoor zonder dat de gebruiker iets van T_EX ziet te genereren. T_EXspell wordt gebruikt om uit (T_EX)-teksten woordenlijsten te genereren ten behoeve van de spellingscontrole. asciiT_EX tenslotte, zet een T_EX-file om in een ASCII-file. Alle genoemde programma's zijn ontwikkeld in MODULA-2.

Recente ontwikkelingen

Inmiddels is de basis van ons systeem redelijk volledig. De meeste uitbreidingen vinden plaats op verzoek van de gebruiker, die ook de β -test voor hun rekening nemen. De macro's zijn dientengevolge redelijk betrouwbaar en worden zo opgezet dat uitbreidingen mogelijk zijn.

In huis wordt momenteel gewerkt aan de aansluiting van T_EX op pakketten die het mogelijk maken, zij het in beperkte mate, te interacteren met teksten. Op dit moment worden DVIWINDO (van Y&Y) en ACROBAT (van ADOBE) ondersteund. Ons inziens vormt T_EX een uitstekend vertrekpunt voor (multi-mediale) documentatiesystemen op de computer.

Natuurlijk blijven er altijd wensen over. De zwakste schakel in ons systeem is op dit moment de output-routine. Naast het perfectioneren van routines ten behoeve van het plaatsen van figuren, zullen ook meer variaties in de layout mogelijk moeten zijn, zoals het plaatsen van figuren en andere tekstelementen op de linkerpagina, terwijl de tekst rechts staat. De hiertoe reeds ontwikkelde routines zullen moeten worden uitontwikkeld.

Zoals hierboven vermeld, is een verplaatsingsmechanisme geïmplementeerd waarmee delen van de tekst kunnen wor-

den gemarkeerd, verborgen en op andere plaatsen opgeroepen. Dit mechanisme zal worden uitgebreid met oproepen op naam en sorteren van verplaatsen blokken op naam, bijvoorbeeld om in de tekst opgenomen definities nog eens in een apart hoofdstuk gesorteerd te kunnen zetten.

Het controleren van instellingen van gebruikers is binnen T_EX tot op zekere hoogte mogelijk. Een eerste versie van een syntax-check is weliswaar geïmplementeerd, maar is onvoldoende robuust en betrouwbaar om daadwerkelijk bruikbaar te zijn. Wellicht dat we op termijn een (uni-verseel) programma ontwikkelen dat controle buiten T_EX mogelijk maakt.

Een vierde wens betreft het vaststellen van de kwaliteit van een tekst in termen van leesbaarheid, structuur en aansluiting op een doelgroep. Vooralsnog bestaan hierover alleen ideeën.

Als gevolg van het feit dat de wijze van adresseren per organisatie kan verschillen zijn speciale macro's ontwikkeld die samenwerken met het programma T_EXadres. Het programma beperkt zich momenteel tot adressen en zal daarom in de toekomst ook geschikt worden gemaakt voor andere toepassingen.

Tot slot

Alle in's en out's van T_EX leren kennen kost veel tijd en experimenteren. Het zelf ontwikkelen van een macro-

pakket, inclusief handleidingen, is daarnaast ook niet in enkele dagen gerealiseerd. Boeken, zoals 'T_EX in practice' van Stephan von Bechtolsheim bevatten weliswaar veel voorbeelden maar missen het overzicht. Het door mij meest geraadpleegde boek is, naast het T_EX-book, dan ook het door Victor Eijkhout geschreven 'T_EX by Topic'.

Niet alle geïnvesteerde tijd betaalt zich terug, afgezien van de leerervaring. Zo kom je er achter dat rekenen in gebroken getallen soms beter niet in T_EX kan worden gedaan en dat het vrij lastig, zo niet onmogelijk is om in T_EX een écht gebruikersvriendelijk systeem te maken. Storender is het als je na veel experimenteren moet concluderen dat een bij jezelf gezochte fout ergens anders ligt: moeilijk te positioneren eps-figuren uit HARDVARD GRAPHICS, verwisselde hoogte en breedte bij HIJAAK, fouten in dvi-drivers enz.

Al met al kan worden geconcludeerd dat de investering in tijd en geld zich in de door T_EX geleverde kwaliteit en flexibiliteit dubbel en dwars terugbetaalt. Het is ook aardig te constateren dat T_EX, dat nu zo'n tien jaar beschikbaar is, nog steeds niet of nauwelijks wordt voorbijgestreefd door andere systemen. En dat terwijl de broncode van T_EX beschikbaar is. Wat anderen brengen als volsterkt nieuw is voor T_EX-gebruikers vaak al vanzelfsprekend. Ons inziens kunnen we met T_EX nog jaren vooruit.

Computerondersteund tekstgebruik

J. Hagen

Onderwijskundig Bureau voor Advies- en Ontwikkelwerk,
Postbus 125,
8000 AC Zwolle

11 januari 1994

Context

Naast opleidingsmateriaal en rapporten, produceert PRAGMA teksten die veelal een functie als naslagwerk vervullen. Hierbij kan men denken aan praktische studieteksten, werkplekinstructies en handleidingen bij machines. De keuze voor \TeX is destijds ingegeven door de hoge typografische kwaliteit, flexibiliteit en onafhankelijkheid van dit systeem.

In toenemende mate wordt informatie aangeboden via een beeldscherm, bijvoorbeeld handboeken en handleidingen. Vaak laat zowel de typografische kwaliteit als de kwaliteit van de weergave op het scherm te wensen over. Tegenover het nadeel van een matig leescomfort staat het voordeel van betere toegankelijkheid en relatief eenvoudige onderhoudbaarheid. De media papier en beeldscherm vullen elkaar dan ook eerder aan dan dat zij elkaar uitsluiten.

Aanleiding

Onlangs hebben wij onze dvi-processoren ingeruild voor DVIPSONE en DVIWINDO, beiden van Y&Y. Wij zijn over deze programma's zeer tevreden, zij het dat DVIWINDO figuren in een lage resolutie toont (preview). In dit programma kunnen met behulp van het `\special`-commando koppelingen tussen tekstdelen worden gelegd. Dat wil zeggen dat het aanklikken van een bepaald tekstdeel resulteert in het gaan naar een ander tekstdeel. Hiermee komt een vorm van computerondersteund tekstgebruik binnen bereik, waarbij een hoge typografische kwaliteit wordt gehandhaafd. Dit was voor ons reden om de mogelijkheden van computerondersteund tekstgebruik nader te onderzoeken.

Eisen

Binnen PRAGMA maken we gebruik van een eigen macropakket. Vanzelfsprekend maakt een referentiemechanisme deel uit van dit pakket. Het is mogelijk te verwijzen naar alle daarvoor in aanmerking komende elementen van een tekst, zoals hoofdstukken, paragrafen, figuren, tabellen, formules, definitives, opsommingen, voetnoten enz.

Er is meer nodig dan een referentiemechanisme om een tekst op de computer toegankelijk te maken. Ook het aanklikken van onderdelen in overzichten, zoals een inhoudsopgave, en ingangen in indices moet resulteren in

het naar een bepaalde positie gaan. Bij voorkeur moet op elke bladzijde naar relevante overzichten kunnen worden gesprongen, bijvoorbeeld vanuit een menu. Ook moeten woorden in de tekst kunnen worden aangeklikt. Tenslotte ligt een koppeling tussen (delen van) figuren en tekst voor de hand. Samengevat dient een interactiemechanisme dus tenminste de volgende functionaliteit te bezitten:

1. springen naar het onderwerp van een verwijzing
2. springen naar een onderdeel van een lijst of een index
3. springen naar uitleg bij een woord
4. springen naar uitleg van (delen van) figuren
5. springen vanuit een menu naar lijsten, hoofdstukken enz.

Eis 1 brengt met zich mee dat de plaats van de verwijzing wordt gemarkeerd. Klikken op *zie figuur 1* moet resulteren in een sprong naar de betreffende figuur.

Voldoen aan eis 2 maakt het nodig de onderdelen van lijsten en ingangen in een index te markeren, zowel in de lijst als in de tekst zelf. Bij een index moet klikken op een paginanummer resulteren in een sprong naar de relevante pagina én plaats in de tekst. De voorkeur gaat uit naar klikken op een paginanummer, omdat bij een ingang meerdere verwijzingen mogelijk zijn. In een inhoudsopgave moet het klikken op een hoofdstuknummer of de tekst leiden tot een sprong naar het betreffende hoofdstuk. Hetzelfde geldt voor overzichten van figuren, tabellen enz. Alle hiervoor noodzakelijke verwijzingen moeten door \TeX (macro's) automatisch worden gegenereerd.

Eis 3 is een variant op een standaard referentiemechanisme en heeft alleen typografische gevolgen: visueel moet duidelijk zijn welke woorden aan te klikken zijn.

Om in een figuur een onderdeel te kunnen aanwijzen (eis 4), moet een deel van het figuur kunnen worden gedefinieerd als 'button'. Het daarvoor gebruikte mechanisme moet nauw samenwerken met zowel de plaatsing van de figuur in de tekst als het schalen van de figuur. Verwijzen moet twee kanten op kunnen: vanuit de tekst naar bijvoorbeeld een onderdeel van een afgebeelde machine of vanuit de tekst over bijvoorbeeld een kenmerk van een gebouw naar een representatief element op een foto. Dit alles kan worden verwezenlijkt door over de figuur een (meegehaalde) onzichtbare matrix te leggen.

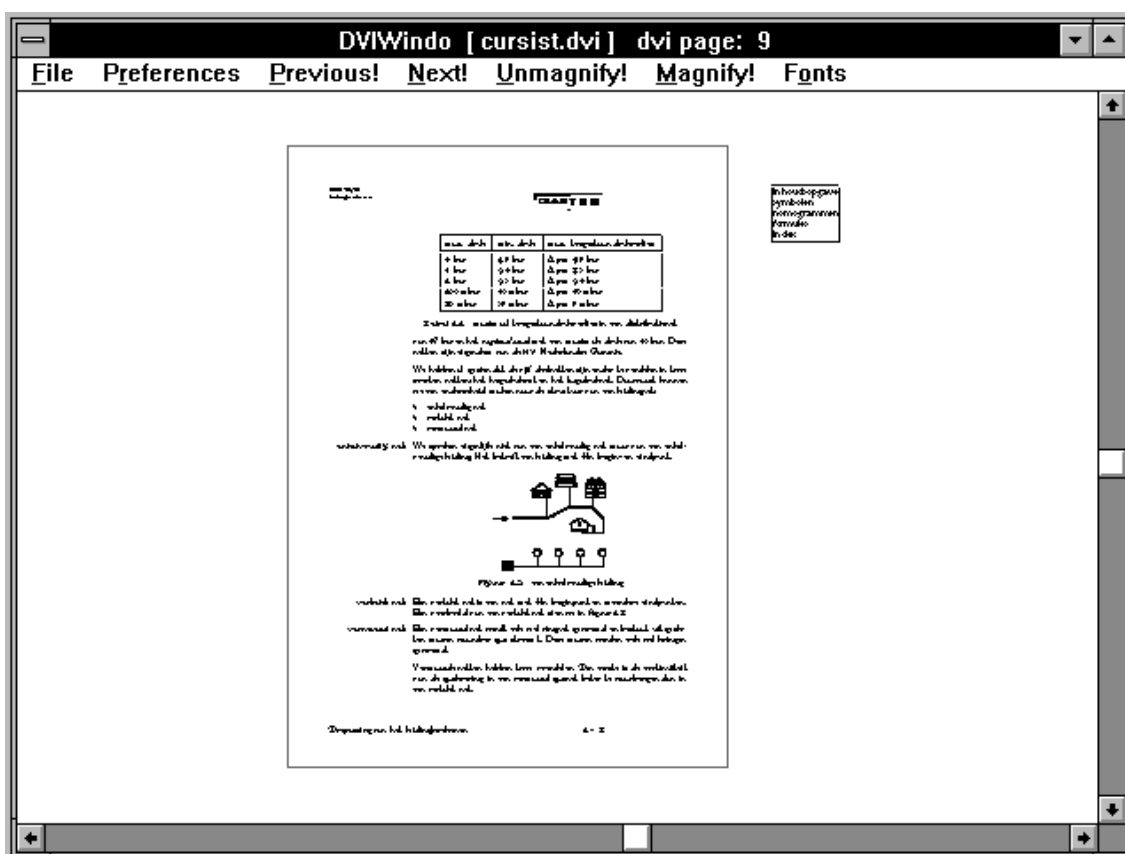
De laatste eis tenslotte, heeft tot gevolg dat in de output-routine aan elke bladzijde een menu wordt toegevoegd met eerder gedefinieerde keuze-items. Deze items kunnen weer door middel van het gebruikelijke referentiemechanisme aan hoofdstukken, paragrafen, bladzijden enz. worden gekoppeld. Hierop aansluitend moeten de voorvieren gebruikte programma's wel de mogelijkheid bieden naar de vorige plaats in de tekst terug te gaan.

Als extra randvoorwaarde kan nog worden genoemd, dat in een tekst zo min mogelijk extra zetcommando's moeten worden toegevoegd. Een tekst moet bij voorkeur zowel

geschikt zijn voor weergave op papier als voor raadplegen op de computer.

Implementatie

De genoemde functionaliteit is binnen PRAGMA inmiddels volledig gerealiseerd. Daarbij worden twee (pre)viewers ondersteund. Het eerder genoemde DVIWINDO stelt de minste eisen, omdat de koppelingen door middel van labels tot stand komen. Het is in dit programma mogelijk de cursor exact op de gevraagde plaats te positioneren. De oplossing is ook relatief goedkoop: de previewer kost slechts enkele honderden gulden.



Figuur 1: Een voorbeeld van de weergave van tekst in dviwindo

In figuur 1 is een voorbeeld van een scherm weergegeven. Rechts van de tekst is een in T_EX aangemaakt menu zichtbaar. Aanklikken van een item in dit menu resulteert in een sprong naar de inhoudsopgave, de index, een overzicht van formules enz.

Uitgaande van het beschikbaar zijn van een volwassen referentiemechanisme, kunnen in DVIWINDO relatief eenvoudig teksten met een, weliswaar beperkte interactiviteit worden geproduceerd. Dit gebeurt met behulp van de commando's:

```
\special{mark: "label"}
```

en

```
\special{button: width height "label"}
```

Met het eerste commando markeren we een plaats in de tekst en met het tweede commando maken we een (onzichtbare) 'button' aan. Klikte men met de muis op deze button, dan wordt naar de plaats met hetzelfde label gesprongen.

Sinds kort brengt ADOBE het programma ACROBAT op de markt. Met dit programma kunnen PostScript-files worden omgezet in het pdf-formaat (portable display format). Files in dit formaat kunnen op verschillende platforms worden bekeken. Door middel van het \special-commando kunnen pdf-commando's in de PostScript-file worden opgenomen:

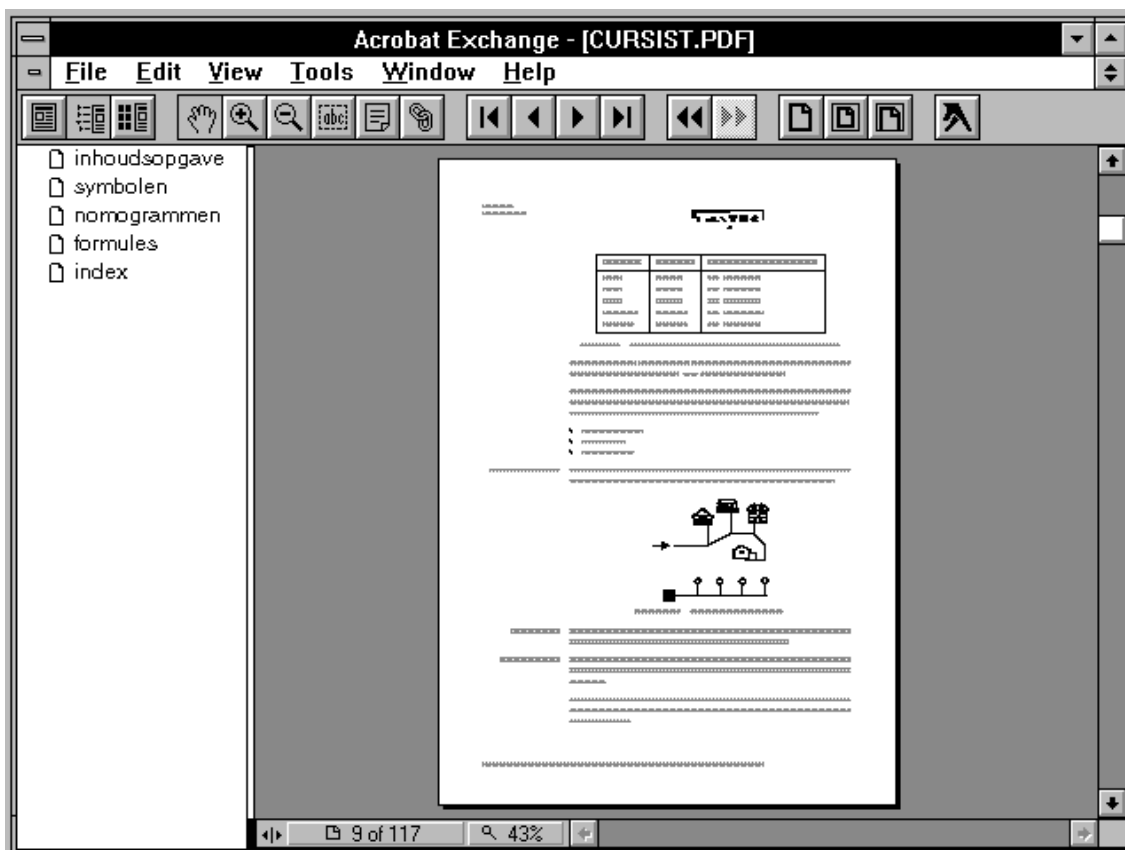
```
\special{postscript [/ ... .. pdf]}
```

Helaas kan alleen naar (delen van) pagina's worden gesprongen. Dit betekent dat bij elke verwijzing naast het

gezette paginanummer het volgnummer moet worden bewaard. Er wordt dus niet gesprongen naar bladzijde 3–5 of VIII maar naar volgnummer 23. Bovendien springt de cursor niet naar de gevraagde plaats in de tekst of lokatie in een figuur. Dit is vanuit $\text{T}_{\text{E}}\text{X}$ niet eenvoudig in waterdichte pdf-aanroepen te vertalen. Het verbaast mij trouwens dat het enkele duizenden guldens kostende ACROBAT niet tenminste het label-mechanisme van DVIWINDO biedt. Overigens gaat men er kennelijk vanuit dat de gebruikers van

het pakket zelf geen software ontwikkelen. De bijgeleverde documentatie is minimaal en de foutmeldingen vallen buiten het niet te vergroten window.

De eerder genoemde menu's kunnen in ACROBAT worden geïmplementeerd met behulp van 'bookmarks'. Deze leveren links op het scherm een verkorte inhoudsopgave op. In figuur 2 is een tekst weergegeven waarbij rechts een menu zichtbaar is met dezelfde keuzemogelijkheden als in figuur 1.



Figuur 2: Een voorbeeld van de weergave van tekst in acrobat

De bovenstaande referentie-mechanismen vreten geheugen, met name uit $\text{T}_{\text{E}}\text{X}$'s stringpool. Een 'big' $\text{T}_{\text{E}}\text{X}$ -implementatie, gecombineerd met macro's die doelmatig gebruik maken van de stringpool, is dan ook noodzakelijk, omdat interactieve documenten meestal omvangrijk zijn. In $\text{emT}_{\text{E}}\text{X}$ kan trouwens de stringpool meer dan voldoende worden vergroot.

Samengevat zijn er dus twee mechanismen naast elkaar nodig om teksten in meerdere (pre)viewpakketten interactief te kunnen bekijken: 'link by label', zoals in DVIWINDO, en 'link by page', zoals in ACROBAT.

Ontwikkelingen

Waar DVIWINDO uitblinkt in eenvoud en snelheid, biedt ACROBAT op termijn meer mogelijkheden. Het is bijvoorbeeld voor lezers mogelijk aantekeningen te maken in de

elektronische tekst. Daarnaast worden figuren in een hoge resolutie weergegeven.

Het zou mooi zijn als in de nabije toekomst in ACROBAT ook bewegende beelden en geluidsfragmenten kunnen worden opgeroepen. In plaats van terug te moeten vallen op voor wat betreft zetwerk matige pakketten, kan dan $\text{T}_{\text{E}}\text{X}$ worden gebruikt als basis, met alle inmiddels bekende voordelen van dien.

Een viewer, uitgaande van dvi-files zou, naast het besproken label-mechanisme, ook de mogelijkheid moeten bieden via `\special's` delen van de tekst te verbergen, dit om geneste menu's te kunnen implementeren. Ook zou een soort `execute program` commando moeten worden ondersteund. Externe programma's kunnen dan bijvoorbeeld bewegende beelden laten zien. Dergelijke mogelijkheden worden op dit moment bij PRAGMA verder onderzocht en ontwikkeld.

Data publishing

Siep Kroonenberg

Economics Department, Groningen University

Abstract

This paper demonstrates how commercial software and \LaTeX can work together in data publishing.

Keywords: database, spreadsheet, table

Introduction

One of (\LaTeX) 's strong suits is automation. Every \LaTeX -user knows about \LaTeX 's features for automatic cross-references and automatic tables of contents. But some people may not realize how convenient \LaTeX can be for publishing data from *commercial* programs. Therefore I present here a couple of real-life examples.

CCSO, the economic research group with which I am associated specializes in the building of econometric models, for forecasting, policy analysis and theoretical exercises. The data for a model typically consist of several hundreds of time series. Now and then we publish these databases, both the data themselves and associated descriptive information.

1 The descriptive database

The descriptive database contains for each time series a description in economic terms and a definition which contains either its source or a definition in terms of other series. A series taken directly from a published source (in this case, the 'CBS Kwartaalrekeningen') might have the following database entry:

```
CNAME      COGRABS
TEXNAME    C_o
DESCRP     Consumption of general
           government at current prices
DEFINITION 1000 * CBS KR 1.3.1 row 5
```

For a constructed series, we might have

```
CNAME      CP
TEXNAME    \dot{C}_p
DESCRP     Growth rate of consumption of
           households at constant prices
DEFINITION 100*(CPKLABS / CPKLABS(-1) - 1)
```

CNAME indicates the name used in the computer models and the spreadsheets; TEXNAME the \TeX code used in papers; DESCRP is the definition in economic terms, and DEFINITION specifies either the source or how it is computed from other series.

We wish to emulate and improve upon a 'native' dBase report, in which each field has its own column, and long fields are wrapped within their column.

There are two problems associated with using the tabular environment for this purpose: first, a tabular environment can't span more than one page, which is not nearly

enough, and second, a tabular environment justifies text even within a narrow column, which can be very ugly.

The first problem is solved by using Theo Jurriëns' `\supertabular` environment, which is defined in the `supertab` style file or package. This environment also allows the specification of a table header which is repeated on every page.

We solve the second problem in a less elegant way: we wrap each long field in a `parbox` of appropriate width:

```
\newcommand{\px}[1]{\parbox[t]{1in}%
  {\raggedright #1}}
\newcommand{\py}[1]{\parbox[t]{1.3in}%
  {\raggedright #1}}
```

The following dBase commands create a file `db94tex.txt` containing a listing in \LaTeX format:

```
set alternate to db94tex
set talk off
set console off
set alternate on
go top
do while .not. eof()
do dbrec
skip
enddo
set alternate off
close alternate
set console on
set talk on
```

This is the procedure `dbrec` that was called above:

```
procedure dbrec
?'$'+trim(texname)+'$ &'
?'\tt '+trim(cname)+' &'
?'\px{'+trim(descrp)+'} &'
?'\py{'+trim(definition)+'} \\'
?
return
```

A fragment of output from these commands:

```
$C_o$ &
\tt COGRABS &
\px{Consumption of general government at
  current prices} &
\py{1000 * CBS KR 1.3.1 row 5} \\'
$\dot{C}_p$ &
\tt CP &
\px{Growth rate of consumption of
  households at constant prices} &
\py{100 * (CPKLABS / CPKLABS(-1) - 1)} \\'
```

The following L^AT_EX code will process the file db94tex.txt (remember to include supertab as a documentstyle option):

```
{
\renewcommand{\arraystretch}{2}
% for extra vertical space between records;
% a pair of braces limits this redefinition
% to the table

\tablehead{& \bf Series & \bf Description & %
\bf Source/definition\\}
\begin{supertabular}%
{@{}p{0.3in}p{0.5in}p{1in}p{1.3in}@{}}
\input db94tex.txt
\end{supertabular}
}
```

And we get:

	Series	Description	Source/definition
C_o	COGRABS	Consumption of general government at current prices	1000 * CBS KR 1.3.1 row 5
\dot{C}_p	CP	Growth rate of consumption of households at constant prices	100 * (CPKLABS / CPKLABS(-1) - 1)

The alternative: tabbing

The example can be easily adapted for a tabbing environment; in fact, that is what we actually did, ignorant of the blessings of supertabular. We even managed to press the page header into service as a repeating table header. But now we know better.

2 The timeseries spreadsheet

Our timeseries are stored in spreadsheets, such as

	A	B	C	D	E
1		BPRGRABS	BPRKLABS	COGRABS	COKLABS
2		$\$B\!P\!R\!$	$\$bpr\!$	$\$C_o\!$	$\$c_o\!$
3	1980Q1	80500	83831	14100	15041
4	1980Q2	86400	87945	17700	18233
5	1980Q3	82000	82501	13700	13842
6	1980Q4	87800	87800	14700	14700
7	1981Q1	84800	84250	15000	15643
8	1981Q2	89700	87945	18500	18780

The first column contains dates; 1981Q1 indicates the first quarter of 1981. The first row contains the names used in the computer model; the second their T_EX equivalents (in

practice, we add the T_EX names only when we are going to print the database; for this, we use the descriptive database discussed above).

Generating T_EX code from such a spreadsheet requires fewer tricks than in the dBase case since we don't have to wrap lines within cells. However, it can't be automated as effectively because, in spite of their macro capabilities, spreadsheets are not really geared towards batch processing. Maybe some day I'll write a Pascal program that does it right.

Meanwhile, we'll let Lotus' print facility generate the output. With some suitable macros (not detailed here), it isn't too much of a job to massage the spreadsheet into shape: we add '&' alignment symbols with a macro, and a column of '\\\new-line symbols by hand. Printing a block to a file timesers.prn we get something like

```
& \dot{C}_p\$\\[1pc]
1980Q1 & 80500 & ... & -4.4301 \\
1980Q2 & 86400 & ... & 0.7075 \\
1980Q3 & 82000 & ... & -0.5809 \\
1980Q4 & 87800 & ... & 3.4624 \\[6pt]
1981Q1 & 84800 & ... & -5.3955 \\
1981Q2 & 89700 & ... & 0.1937 \\
```

The lines

```
\begin{tabular}{@{}lrrrrr@{}}
\input timesers.prn
\end{tabular}
```

produce the table below:

	BPR	bpr	C_o	c_o	\dot{C}_p
1980Q1	80500	83831	14100	15041	-4.4301
1980Q2	86400	87945	17700	18233	0.7075
1980Q3	82000	82501	13700	13842	-0.5809
1980Q4	87800	87800	14700	14700	3.4624
1981Q1	84800	84250	15000	15643	-5.3955
1981Q2	89700	87945	18500	18780	0.1937

3 Caveats

Watch out for characters in the database or spreadsheet which are also markup codes for T_EX, such as percent- or ampersand characters. In spreadsheets, texts should fit within their columns, or they will be truncated without warning. Also, if the block to be printed exceeds the declared linelength of the printfile, the table will be split horizontally.

Table design

Siep Kroonenberg

L^AT_EX users generally seem unaware of current ideas on table design. The following table is a typical L^AT_EX production:

L ^A T _E X table design		
	1991	1992
Unemployment ($\times 1000$)	500	600
Balance of Payments (10^9 Hfl)	24	25

In a professionally-designed publication, the above table would probably look more like this:

Common sense table design

	1991	1992
Unemployment ($\times 1000$)	500	600
Balance of Payments (10^9 Hfl)	24	25

If you read a book on typography, e.g. [1] or [2]: you'll find that they use rules and boxes with far more restraint, and rely more on white space and variation in typefaces for organization.

The table examples in [3] were (I hope) merely intended to demonstrate techniques. However, their style was almost unanimously adopted by L^AT_EX users.

So I think that some design education is in order. I am not a design professional. However, many people never even think about table design; so if I set them thinking and they start reading books on typography by real professionals then this paper has served its purpose.

Note. This is not meant to be a technical exposition. [4] and [3] tell you most of the technical things you need to know. All the same, I have indicated here and there with what codes or constructs you might accomplish certain effects.

1 Basics of table design

A table should present its information as clearly as possible. Typographic means to organize this information includes rules, white space, choice of typefaces and appropriate headings and captions. But if a feature doesn't help to make a table clearer, it had better be left out.

Macroeconomic memoranda

1.	Karl Lausche, Maria Vader, Theo Zernike	Money illusion and savings illusion; an illusionistic look on neo-Hegelian monetary theory
2.	Hendrik Kooyker, Johan Zonderlink	BIGTHUMB, a software package for handling missing and politically incorrect data
3.	Anneke Draaijer	Consumer behavior, expectation formation and the long-term economic effects of risk-aversion

1.1 Rules and boxes

Rules have their uses. They can emphasize headings. They can also separate different items and unite the several data for one item, as in the table above. Vertical rules, as in the table below, would have the opposite effect and would be no help at all in making the table easier to read.

Macroeconomic memoranda

1.	Karl Lausche, Maria Vader, Theo Zernike	Money illusion and savings illusion; an illusionistic look on neo-Hegelian monetary theory
2.	Hendrik Kooyker, Johan Zonderlink	BIGTHUMB, a software package for handling missing and politically incorrect data
3.	Anneke Draaijer	Consumer behavior, expectation formation and the long-term economic effects of risk-aversion

But even in the earlier example one might wonder whether white space wouldn't have been more effective than rules.

A table may also be boxed to set it off from the surrounding text. But L^AT_EX users normally don't go through the trouble of wrapping text around tables and figures; therefore, there is little reason to box in a table.

In all cases, there should be enough space between rules and text. A rule too close to text interferes with readability and makes the text look cramped.

An alternative to rules or boxes is a shaded background, preferably in a second color. This is not supported by L^AT_EX as far as I know, although with PostScript some tricks are possible (see e.g. [4] section 11.6). This formatting device requires high output quality in order to look good.

1.2 Alignment and justification

A column of text labels can be left- or right-aligned, or centered. If the table has any length at all, a centered column can easily look sloppy. With left- or right-alignment there is at least one straight edge to give the column structure. Think twice before centering a column in a longer table.

A column of figures is usually decimally aligned (see under section 3 for some technical issues). If the figures are unrelated, you may consider right- or left-alignment instead.

Don't justify text inside a narrow column or you'll end up with large distracting holes between words. This is easier said than done, but see under Section 3.

1.3 Headings

Headings may get added emphasis by setting them bold, italic, at a larger point size or in a different typeface. Don't go overboard, though. The heading of a centered or decimally-aligned column may need some manual adjustment.

2 An example

We illustrate some of these points with a before-and-after example, see below. It is sufficiently complex to illustrate a number of points; I am not implying that it is any worse than other L^AT_EX tables I have seen. The 'before' table is a L^AT_EX remake of a table from [5]. At an earlier occasion, it has been used as a demonstration of L^AT_EX's table-making capabilities.

Example: before and after

Economic forecasts for 1992

variabele	forecast	
	Grecon	CPB (MEV '92)
	% mutations w.r.t. 1991	
real consumption (<i>c</i>)	1.1	1.25
price index consumption (<i>p_c</i>)	2.6	3.25
real investments (<i>i_m</i>)	1.4	-2.5
export price index (<i>p_b</i>)	3.5	3.25
real import of goods (<i>m</i>)	4.1	3
real output of goods (<i>v'</i>)	2.5	2.1 ^a
real domestic production (<i>b_{pr}</i>)	1.4	1.6
private employment (<i>a</i>)	0.32	0
wage rate (<i>l</i>)	4.0	4
government income (%) from output of goods (<i>iso'</i>)	-0.1	- ^b
	absolute quantities	
unemployment (×1000 persons)	510 ^c	525
balance of payments (10 ⁹ Hfl)	24.4	25.0

^aThe quantities *v'* and *b_{pr}* aren't given as such by the CPB. The CPB data presented here are computed using their GRECON definitional equations. For details, see appendix D.

^bNot available.

^cNot a model outcome: see text in par. 3.1 en 3.2.

The example table contains footnotes; therefore it is enclosed in a minipage environment.

2.1 Rules

The most conspicuous shortcoming of the 'before' example is the tight spacing between horizontal lines and text. I am not aware of a parameter which controls this distance; however, the '\l' command takes an optional length parameter, also in a \tabular environment.

In this case, as in most cases, the vertical rules are better left out. It is advisable to begin and end the column specification with @{}:

```
\begin{tabular}{@{}l@{}r@{}l@{}l@{}}
```

Without vertical rules, no white space needs to be reserved at the left- and righthand sides.

Actually, I used a tabular* environment, which allowed me to set the width to \linewidth: exactly the width of the minipage.

Another unfortunate detail is the footnote rule next to the bottom rule. I solved this by dropping the bottom rule. Also, I redefined in a separate style file several aspects of minipage footnotes: among others, the footnote rule now stretches across the width of the minipage.

The rule under the title is not part of the \tabular-environment, but is constructed as a '\rule'-rule. This made it easy to give it a custom thickness. Again, the length was set to \linewidth.

Economic forecasts for 1992

	Grecon	CPB ^a
<i>mutations w.r.t. 1991</i>		
real consumption (<i>c</i>)	1.1	1.25
price index consumption (<i>p_c</i>)	2.6	3.25
real investments (<i>i_m</i>)	1.4	-2.5
export price index (<i>p_b</i>)	3.5	3.25
real import of goods (<i>m</i>)	4.1	3
real output of goods (<i>v'</i>)	2.5	2.1 ^b
real domestic production (<i>b_{pr}</i>)	1.4	1.6
private employment (<i>a</i>)	0.32	0
wage rate (<i>l</i>)	4.0	4
government income (%) from output of goods (<i>iso'</i>)	-0.1	- ^c
<i>absolute quantities</i>		
unemployment (×1000)	510 ^d	525
balance of payments (10 ⁹ Hfl)	24.4	25.0

a. MEV '92

b. The quantities *v'* and *b_{pr}* aren't given as such by the CPB. The CPB data presented here are computed using their GRECON definitional equations. For details, see appendix D.

c. Not available

d. Not a model outcome: see text in par. 3.1 en 3.2.

2.2 Headings

As to the various headings: the word *forecast* repeated information from the table header and was dropped. The word *variable* could also safely be omitted.

Aligning the Grecon- and CPB headings at the bottom instead of the top would have been an improvement, but moving the text ‘*MEV ’92*’ to a footnote was even better. Their horizontal positioning was adjusted by hand, adding ‘~’ here and there.

The ‘*% mutations...*’ and ‘*absolute quantities*’ headings looked rather jarring in the figures columns, and were moved to the left column.

2.3 Fonts

Sans serif faces are especially appropriate for tabular material. At small sizes serifed faces easily look fussy, especially if the output quality is not top notch. Sans serif faces suffer much less from scaling down. A sans serif face also helps to set off the table from the surrounding text.

Several sizes and weights are used (typographers talk about an italic *weight*; the T_EX community should realize that they entertain rather off-beat ideas about font families). And hyphens are replaced by proper minus-signs.

3 Technical issues

Some things in L^AT_EX are harder than they should be. Two notorious examples are table-related: aligning a column of figures on the decimal point, and setting text in a table cell ragged right.

3.1 Decimal alignment

There are at least three ways in L^AT_EX to accomplish decimal alignment:

- If all numbers have the same number of digits after the decimal point, decimal alignment coincides with right alignment, since in most fonts all digits have the same width.
- Split the numbers right before the decimal point, i.e. put an ampersand ‘&’ before the decimal point (or after the number, if it has none). The column formatting for the resulting two columns should be `r@{ }l`: right-align the part before the decimal point, left-align the remainder, and put no white space in between.
- Use the `dcolumn` package by David Carlisle. This is documented in [4] section 5.5.1.

3.2 Ragged right justification

You may have noticed that `\raggedright` simply doesn’t work in a tabular environment. Again, let me suggest a couple of brute-force workarounds.

- Divide the text manually between rows. Of course, this is practical only in very simple cases.
- Put a parbox around the text, e.g.


```
\parbox{1in}{\raggedright text...}
```

 This is simple enough, but not very elegant since

it involves specifying column widths outside the `\begin{tabular}` command.

For a more sophisticated solution, see [4] section 5.3.1, ‘*Typesetting Narrow Columns*.’ As in the last of the above two workarounds, it adds code to make `\raggedright` operational again.

4 The trouble with L^AT_EX

It took me a lot of time to prepare the examples in this paper. Even the standard L^AT_EX `\tabular` environment has plenty of quirks, and extension packages such as `array` or `tabularx` only add to them. Too often, it was a matter of trial and error what would work and what wouldn’t, and that might depend on the package used. In the end I didn’t use any of the table extension packages for this paper.

In L^AT_EX, some aspects of layout and typography can be controlled by changing a few parameters or by replacing some simple code out of a style file. But there are quite a few rough spots: sometimes the code is too cryptic for easy modification and sometimes the code is not in the style file at all. When typesetting tables one tends to run into such rough spots.

Besides L^AT_EX, I use high-end wordprocessors and low-end desktop publishing software. I am exceedingly frustrated that simple things that you just do in a commercial program, require hours or days of study and experimentation in L^AT_EX.

Still, L^AT_EX can’t be beat (yet) for long documents or for automation. It remains robust and efficient whatever the size and complexity of the job. So I keep using it for certain types of work.

I hope that (L^A)T_EX developers are seriously addressing L^AT_EX’s shortcomings. What is really needed is a more accessible basic L^AT_EX system, which doesn’t require wizardry to tailor to one’s own preferences, and which can put an end to the current proliferation of style files to patch up its defects.

Finally I want to mention that [4] was a great help in preparing this paper, even though the solutions proposed there didn’t always work out.

References

- [1] Treebus, K. F. *Tekstwijzer*. SDU 1982.
- [2] McLean, Ruari. *Typography*. Thames and Hudson 1980.
- [3] Lamport, Leslie. *L^AT_EX, A Document Preparation System*. Addison-Wesley 1986.
- [4] Goossens, Michel, Frank Mittelbach, Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley 1994.
- [5] Dietzenbacher, H.W.A., W. Voorhoeve. *Het model GRECON 91-D. Septembervoorspellingen voor 1992*. Onderzoeksmemorandum no. 450. Economics Department, Groningen University 1991.

BLUe's Transparencies— to finish up with

Kees van der Laan

Abstract

T_EX macros for formatting transparencies are provided. The macros have been designed with reusability of earlier formatted document parts in mind, with a wink to maintenance of the transparency set.

Keywords: Education, foils, formatting, generic style, slides, T_EX, (black-and-white) transparencies.

1 Introduction

This is the second paper¹ in the series where I select a BLUe-way² of formatting with T_EX from the multitude of confusing possibilities.

When I started to use T_EX in the mid-80-ies my main use was formatting and typesetting reports and—derived from those—preparing and making transparencies. Not hindered by any experience—as newbie T_EXie—I modified L^AT_EX's `report.sty`. This work emerged in `trspar.sty`. It did cost me a lot of time, compensated by that I learned something from L^AT_EX's coding style. I released this preliminary and hardly documented version on the TEX-NL network—no comments nor users :(—and have used this tool for my presentations, ever since.

Why `trspar.sty`? Of course I was aware of SliT_EX, but rejected it because it

- was not just another style, the (outer) markup had to be adapted
- was restricted to L^AT_EX
- did not provide for headers or footers.

Since then I have among others heard of

- FoilT_EX (Hafner)
- Seminar (Rahtz and Goossens), and
- Voyager (Jet Propulsion Lab).

Sensible packages to start with, but my use is different. I don't need colours other than for background, now and in the foreseeable future. That is, basically I use two 'colours': text and background. When the foils are made by Xerox copying, I can just copy them on the appropriate coloured backgrounds. (Or take my BLUe background— or whatever colour is suited for the occasion— with me.) Too many colours make a transparency like a fancy-fair. Sometimes appropriate and fun. For my work I don't need that, generally.

Let us consider classes of transparency sets. I know of the following kind of presentations, where transparencies are often used

- business, with roughly a dozen or so words per transparency and a pie-chart now and then. Much perfection in there and reused frequently.
- scientific, mostly single-shot presentations and therefore hardly reused (no maintenance).
- courses, a large set, typically a hundred or so, with accompanying notes and quite some maintenance involved.

From the above my conclusion is that we need tools for perfect and fancy transparencies to assist business-like presentations, and tools which can handle the maintenance aspects gracefully.

The Seminar tool is L^AT_EX-based, and does not pay special attention to the maintenance aspects. For business presentations it might do.

What is the problem? Why not just use plain with some suitable `\magnification`, or document `trspar.sty` and that is it? Well, . . . the problems are twofold. First the preparation of transparencies is an art too.³ And, second there is the selection problem, what to use from what is available, or develop your own style from. To my experience the latter is important especially in relation with the art issues. Once the tool is personalized one can go for working like a craftsman. Requirements are: simplicity and extensibility.

Essentially, I have used plain with some magnification and some fancy headings. So this is similar to van Oostrum's Fancyheadings, modulo some syntactic sugar.

Furthermore, the coding of `trspar.sty` is a too complex inheritance from L^AT_EX, certainly in relation to its task. (I had to modify several L^AT_EX files, not only `report.sty`.)

¹ The first paper was BLUe's Bibliography, MAPS 93.2. The third is BLUe's Verbatim, MAPS 94.1. Projected are BLUe's Pictures and BLUe's Format.

² BLU stands for Ben Lee User of the T_EXbook fame. BLUe is its cousin which I adopted. The idea is that like in Math the basics are explained to jump off from, to develop your own way.

³ I'm much surprised that I don't use `\subhead` so often, but instead `\shtitle`.

Second, I like a simple, generic and flexible tool which can be adapted easily, be it extended or customised to (L^A)T_EX.

Third, I like a tolerant tool which allows reuse of already 'formatted in the small' document elements. I assume that for markup in the small plain T_EX is used or macros from a general toolbox, which combine easily with plain, let us say which form an orthogonal extension.

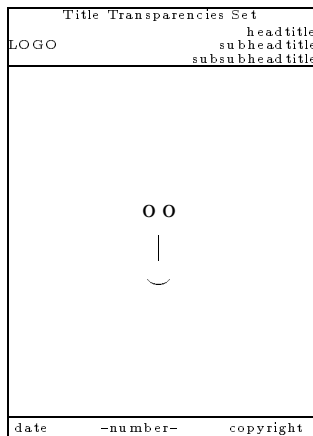
My envisioned use is that people will start from a report or so, and strip off all the 'redundancy' until the main document elements for transparencies remain.⁴ Be it a table, some math or graphics, already marked up in the small and *simply reused*. I had this on my mind ever since I started in the computer-assisted document preparation area. I labelled it from report to transparency.⁵

Notations and definitions. Generic means that the user markup remains invariant.

2 Design

We have two issues: the regular transparencies (pages), and the title page. I consider the latter the exception.

Layout regular transparency. I like to distinguish a header, page proper, and the footer.



The header part consists of

- the title of the set—positioned at the top, centered
- logo—positioned left—and
- context information—positioned right to three levels—reflecting the head, subhead and subsubhead titles, if any.

There is much similarity in spirit with the running head of a book.

The page contents has the structuring commands `\head`, `\subhead`, and `\subsubhead`. For the markup in the small, plain T_EX or any compatible extension can be used.

Footer. I chose for the footer the date positioned left, the copyright positioned right and the transparency 'num-

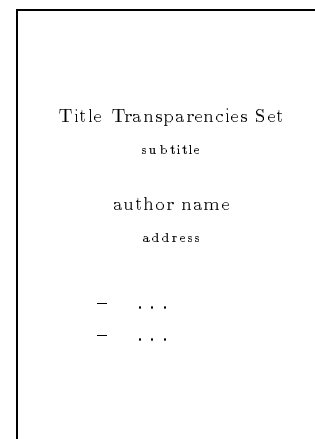
ber' centered. The header and footer are separated by an `\hrule` from the contents proper.

Practice dictated also something like `\continue`, with function that the current transparency will be continued on the next.

Corrections and additions. For a large set this means to prepare these pages separately. In order to do so the environment parameters must be appropriately filled—the titles of the set and the various heads—and it needs its own (extension) date, and page number suffixed by a, b, . . . See Appendix C for an example.

I also divide a large collection into logical smaller sets with numbering relative to each set, prefixed by a characterization symbol.

Layout title transparency. I like to distinguish a title, a subtitle, author name and address, and a list of main items to be treated.



3 Implementation

There are three main issues: fonts, regular transparencies, and the title page.

Fonts. I chose not to use too many fonts. This makes it easier to customize with fonts of your choice.⁶

The benefits which come with PostScript—especially the many professional fonts—are not special for the making of transparencies, and therefore I refrained from those 'specials' for the moment.

Title page. For the data of the title page I adopted the toks variables: `\title`, `\stitle`, `\author`, `\address`, `\contents`. As can be seen from this I also typeset a table of contents below the title and author information, that is a reasonable list of the main issues to be treated. The user is encouraged to supply in the contents toks variable some `\bitem-s`. The title page is set by `\maketitle` invoked by the (first) `\head`.

⁴I know that people involved in education start the other way round and obey Gagné's principles.

⁵The title of an early-80-ies RC-RUG report of mine, called (in Dutch) 'Van rapport naar transparant.' Another approach is to include 'transparencies' as 'illustrative summaries' in the article.

⁶Perhaps when I'm more mature with METAFONT and PostScript I might have a strong opinion on these matters.

Regular transparencies. Here we have two aspects: what is needed as markup commands, and what is done automatically with respect to the header and footer.

- the markup commands are `\head`, `\subhead`, and `\subsubhead`, all with the title as argument. The argument must be terminated by a blank line (or a `\par`) in the \TeX book spirit.⁷ For the markup in the small just plain \TeX commands can be used.
- the header and footer are set automatically, with information distilled from the structure commands and the title. A (user-supplied) logo is set too.

Summary of commands

Token vars

<code>\address</code>	<code>\identification</code>
<code>\author</code>	<code>\shtitle</code>
<code>\contents</code>	<code>\sshtitle</code>
<code>\crownr</code>	<code>\stitle</code>
<code>\htitle</code>	<code>\title</code>

Definitions

```
\continue{...}
\head#1\par{...}
\maketitle{...}
\nxttrs{...}
\regularhead#1\par{...}
\subhead#1\par{...}
\subsubhead#1\par{...}
```

Conclusion

The macros obey the adage: prototype, revise and rewrite. `trspar.sty` can be seen as the prototype. `trs.tex` is simpler, more flexible, faster, and uses less resources.

References

The \TeX book and the \LaTeX book are not explicitly mentioned, because I assume that these are omni-present.

- [1] Goossens M, S.P.Q. Rahtz (1994): Colour slides with \LaTeX and Seminar. Baskerville, 4,1, 8–12.
- [2] Hafner J. (1992): Foil \TeX , a \LaTeX -like system for typesetting foils. *TUGboat* 13, no. (3), 347–356.
- [3] Hemel J., C.G. van der Laan, R.C.G.M. Lauwerier (1989): Het maken van transparanten voor overhead-projectie.RC-Rapport 23. (In this report a set of 4 characteristic transparencies—a drawing, a bar-chart, a pie-chart, a list of items with some formulas—is selected, and the aspects which come along when making these by hand, by Wordperfect, or by \LaTeX , are discussed. The results are shown and the needed amount of work for each approach is summarized among others.)
- [4] Hout J.F.M.J., M.J.A. Mirande, E.B. Smuling (1981): Geven van hoorcolleges. Aula 807. (Apart from the useful survey aspects it supplies also references to Gagné's work.)
- [5] Oostrum P van (1989): Fancyheadings. From the fileserver.

⁷There is a great variety with respect to parameter termination: just a parameter within curly braces, or `\end...` terminators, or enclosed within `*-s` (TUGboat), or `...`. Despite the arbitrariness I favour especially here Knuth's convention to terminate the parameter by a blank line—or a `\par`—because it is close to natural markup: heads take a line by themselves. No run-in text.

Appendix A: The file trs.tex

A sample example is in Appendix B.

```
%trs.tex Febr. 1994 Version 1.0
%Author: C.G. van der Laan, Hunzeweg 57, 9893PB
% Garnwerd, The Netherlands
% 05941-1525, cgl@risc1.rug.nl.
%Purpose: Preparing transparencies from
% TeX marked up copy.
%
1 \font\titelfont=cmr12 scaled1440\relax
2 \font\shfont=cmr12 scaled1200\relax
3 \font\sshfont=cmr12
4 \magnification1728 %2074
5 \font\ffont=cmr7
6 \def\large{\shfont\baselineskip3.25ex}
7 \def\small{\ffont\baselineskip3.5ex}
8 %
11 %Customize what is set by \magnification
12 \hsize=17.5truecm \vsize=19truecm
13 \hoffset-1truecm\voffset3.5truecm
14 %
21 \newtoks\identification
22 \identification={\today}
23 \newtoks\crownr
24 \newtoks\title
25 \newtoks\stitle
26 \newtoks\contents
27 \newtoks\author
28 \newtoks\address
29 \newtoks\htitle
30 \newtoks\shtitle
31 \newtoks\sshtitle
32 %
101 \def\head#1\par{\maketitle\vfil\ejct}
102 \htitle={#1}\shtitle={}\sshtitle={}
103 \centerline{\large#1\quad}\smallskip
104 \let\head\regularhead\pageno=1
105 %
106 \def\regularhead#1\par{\vfil\ejct
107 \htitle={#1}\shtitle={}\sshtitle={}
108 \centerline{\large#1\quad}\smallskip}
109 %
110 \def\subhead#1\par{\shtitle={#1}\sshtitle={}%
111 {\large\goodbreak\medbreak\noindent#1%
112 \smallskip\nobreak}}
113 %
114 \def\subsubhead#1\par{\sshtitle={#1}%
115 {\sshfont\medbreak\goodbreak\noindent#1%
116 \smallskip\nobreak}}
117 %
118 \def\nxttrs{\vfil\ejct}
119 \def\continue{\vfil
120 \rightline{\dots continued}\ejct}
121 %
201 \headline={\line{\vbox to0pt{\vss
202 \centerline{\large\strut\the\title}
203 \vskip2ex
204 \line{\logo\hfill\vbox{%
205 \llap{\strut\the\htitle}
206 \llap{\strut\the\shtitle}
207 \llap{\strut\the\sshtitle}}}}
208 \vskiplex
209 \hrule width\hsiz height1pt
210 }}%end hbox
211 }%end headline
212 %
251 \def\thepage{\folio}
252 %
253 \footline={\vbox to 0pt{%
```

254	\hrule width\hsize heightlpt	\\.....	802
255	\vskip.5ex	\bitem.....	803
256	\hbox to\hsize{%	\today.....	804
257	\rlap{\small\the\identification}%	Personalisation	
258	\small\hfil--\thepage--\hfil	\crown.....	902
259	\llap{\small\copyright\the\crown}%	\author.....	903
260	}\vss}%end vbox	\address.....	904
261	}%end footline	\calx.....	908
262	%	\logo.....	909
301	\def\maketitle{\footline={}\headline={}	%	
302	\bigskip	History of changes	
303	\centerline{\titlefont\the\title}	Feb 1994 Release Version 1.0	
304	\smallskip	d	
305	\centerline{\the\stitle}		
306	\bigskip		
307	\centerline{\large\the\author}		
308	\smallskip		
309	\the\address		
310	\vskip7ex		
311	\centerline{\vbox{\hsize.5\hsize		
312	\large\baselineskip=4ex		
313	\the\contents}}%		
314	}%end maketitle		
315	%		
801	%Auxiliaries		
802	\def\\{\hfil\break}		
803	\def\bitem{\item{\\$bullet\$}}		
804	\def\today{\ifcase\month\or Jan\or Feb\or		
805	March\or April\or May\or June\or July\or		
806	Aug\or Sept\or Oct\or Nov\or Dec\fi		
807	\ \number\year}		
808	%		
901	%Personalisation		
902	\crown={cgl}		
903	\author={Kees van der Laan}		
904	\address={%		
905	\centerline{Hunzeweg 57, 9893PB, Garnwerd}		
906	\centerline{The Netherlands, 05941-1525}		
907	\centerline{cgl@risc1.rug.nl}}		
908	\font\calx=cmsyl10 scaled \magstep2		
909	\def\logo{\vbox to 0pt{\vss		
910	\rlap{\calx N}ederlandstalige}		
911	\rlap{\hskip1em{\calx T}\kern-.2ex		
912	\raise-.5ex\hbox{E}\kern.1exX}		
913	\rlap{\hskip2em{\calx G}ebruikersgroep}%		
914	}}		
915	\overfullrule0pt		
916	\endinput		
	Contents		
	Fonts.....		1
	Newtoks.....		21
	Mark up		
	\head.....		101
	\regularhead.....		106
	\subhead.....		110
	\subsubhead.....		114
	\nxttrs.....		118
	\continue.....		119
	Header and footer		
	\headline.....		201
	\thepage.....		251
	\footline.....		253
	Title matter		
	\maketitle.....		301
	Auxiliaries		

Appendix B: Sample copy

```

\title={Manmac BLUes}
\stitle={---how to typeset a book
via \TeX---}
\contents={\bitem User's guide
\bitem Coding}
%
\head Why?\par
To return to the roots...
%
\shthead{Contents}%Extensive
\bitem Font definition
...
%
\head User's Guide\par
\subhead Chapter markup\par
\subsubhead Structure\par
\def\thepage{U\folio}\pageno=1
...
%copy proper
\nxttrs %Next transparency
\subsubhead Markup paragraphs\par
...
\continue%Continuation on next slide
...
\bye

```

Appendix C: Maintenance

This appendix is in a certain sense artificial. Introduced by the automatisms which come with my choices for the 'header'. If this format and header is adopted the following might be helpful for modifying or extending single transparencies.

```

%Additional/corrections
\nxttrs
%Context
\title={Literate Programming}
\hthead={Tools}
%Suppress title page via
%\let\head\regularhead
%Footer mods
\identification={feb94}
\def\thepage{13a}
%
\subhead FWEB\par
This LP tool is developed by John Krommes
\bye

```

BLUe's Bibliography— selective loading

Kees van der Laan

Abstract

In addition to the earlier method of maintaining a database in \TeX , it is shown how to load the entries selectively, without modifying the entries of the database

Keywords: Literature database, bibliography, plain \TeX , macro writing, education.

1 Introduction

As proposed in my BLUe's Bibliography paper, I like to supply each entry in a (near) natural way, as suggested by the following example

```
Knuth D.E (1984):
The \TeX book.
Addison-Wesley.
ISBN 0-201-13447-0 (hard cover)
ISBN 0-201-13448-9 (soft cover).
(For the right printing look for
\cs{language}, or \cs{emergystretch}
in the index.)
```

which obeys the syntax

```
<name part> (<date part>):
<title part>
<rest>
```

In my use of the above I encountered no problems. The question is whether something easy can be done about the redundancy of loading all entries.

2 Selecting entries.

To remind the reader this section has been copied from the earlier article.

For selecting I made use of the active list separator \ls , with context dependent meaning. When the database file consists of the entries

```
\def\knuthded
{Knuth D.E (1984):
The \TeX book. \aw.
ISBN 0-201-13447-0 (hard cover),
0-201-13448-9 (soft cover).
(For the right printing look for
\cs{language}, or \cs{emergystretch}
in the index.)}
\def\knuthdeg
{Knuth D.E (1986):
The \mf book. \aw.
ISBN 0-201-13445-4 (hard cover).}
```

then for selecting them all, supply

```
\ls\knuthded \ls\knuthdeg
```

Selective loading.

The above proposal loads all the definitions from the database file in \TeX 's memory. Knuth in *The \TeX book*, Appendix D, p. 384, suggests to load only the definitions needed. The critical memory component—if any¹ with a reasonable database of entries—are the 'words of memory.' How much, and which, memory is used, can be requested via $\tracingstats=1$.

Of course there is truth in Knuth's suggestion and I decided to do some experiments for the cases

- no references at all
- insert those needed manually in the copy
- load selectively
- load all definitions.

The 'words of memory' used with the file of my original bibliography article as test are²

<i>no refs</i>	<i>insert man</i>	<i>load selectively</i>	<i>load all</i>
21.5	22.5	27	51.5
–	1	5.5	30

As expected, there is some overhead in the selective loading process. The gains compared with loading all definitions is indeed significant.

I did not like the idea of modifying the database for flexibility reasons. Therefore, I used the following trickery for selective loading, with the syntax of the database elements given earlier and based on \def as the name of a macro.

```
\let\x\def%or any name for x
\x\def#1{\ifx#1\undefined\ea\gobble
\else \ea\x\ea#1\fi}
\input lit.dat
\let\def\x
```

As its name suggests \gobble eats its argument.

References

The \TeX book, and

- [1] Laan C.G van der (1993): BLUe's Bibliography—a generic approach. MAPS 93.2, 205–210.

¹I have typeset reasonable large articles—an odd thirty pages with some 15 references out of 250—with as format `tugboat.sty`, without memory problems of yet. In my opinion it is not a problem at all on a reasonable 'intermediate' \TeX system, because one can always include the references manually in the pen-ultimate proofs, and work in the meantime with labels or some other mnemonic aid. Remember that the entries have been proofed earlier.

²In units of kbytes. The second data row contains the values corrected for the no references situation, that is 21.5 kbyte is subtracted.

BLUe's Verbatim—the selection

Kees van der Laan

Abstract

A suite of macros for verbatim mode has been provided, which are simple, concise and flexible. The functionalities are: verbatim text, and file verbatim inclusion, with 'options' for numbering and enabling of metacode.

Options can be supplied via the toks variable `\thisverbatim`, and globally via `\everyverbatim`.

The place within context can be handled via the defs `\preverbatim` and `\postverbatim`.

The macros are context independent and can be used at the inner level with Any \TeX .

Keywords: Education, escape character, formatting, generic style, semi-transparent, \TeX , verbatim mode.

1 Introduction

This is the third paper¹ in the series where I select a BLUe-way of formatting with \TeX from the multitude of confusing possibilities. In the previous papers I have addressed the audience. To be honest the real reason is that the articles function for me as a mental aid, a summary of what I have once understood.

When writing about (\LaTeX) in \TeX verbatims are needed. By verbatim text I mean that what is keyboarded is formatted in teletype font with spaces and end-of-lines obeyed. The special characters of \TeX are set with catcode 12. The \TeX book provides in Appendix D.3 Verbatim listing

- `\listing`, 380, for file verbatim with line numbers added in the typeset result
- `\verbatim`, 381, for inline verbatim, with an alternative coding on page 382
- `\beginverbatim ... \endverbatim`, 382, for typesetting everything in between in verbatim, except for `\endverbatim`, the termination tag
- typesetting 'tabs verbatim (that is variable number of spaces),' really advanced.

\LaTeX provides the functionalities

- `\verb<char><verbatim text><char>`, for inline verbatim, and
- `\beginverbatim ... \endverbatim`, for a displayed result.

TUGboat.sty provides a fancy set of macros which can't be used easily in other contexts. The functionalities are rich, however. The best and most complete I have seen.² See my TUGboat BLUes, for an elaborate discussion.

The aim of this paper is to select a simple, flexible, practical, and generic suite.

2 What is the problem?

The problem is apparently the incompleteness of the available macros for practical purposes. Be it that options are lacking, that there is no file verbatim, or that the encoding is dependent upon the general mechanisms of the environment.

For BLUe the problem is the selection of a practical and general suite, which is simple, intelligible, flexible, and generic. With the latter I mean that it can be used within any context, be it plain, manmac, \LaTeX , or $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$, to name but the most notorious flavours of \TeX .

In keeping it simple I sacrificed uniform markup conventions. I chose to start verbatims with `\verbatim` but to end the *mode* via `<escapechar>verbatim`. This looks strange. I think it does not have to be perceived as such, especially when we realize that we can assume some \TeX knowledge with authors who write in \TeX about (\LaTeX) , and who will appreciate 'exceptions to the rule.'

3 Design

Functionally, I needed in practice³

```
%1. To handle via default ! escape char
% -- other fonts
% -- footnotes
% -- emc (enable metacode)
\thisverbatim={\emc
  \def\ftxt{Footnote text typed in
             on more than one line.}}
\verbatim
Some <meta code> and
blah, blah, ... !it
Now text in italics!tt
and back again in tt
footnote!footnote*!bgroup !ftxt!egroup
!endverbatim
```

¹ The first two papers are BLUe's Bibliography, MAPS 93.2, and BLUe's Transparencies, MAPS 94.1. Projected are BLUe's Pictures and BLUe's Format. BLU stands for Ben Lee User of the \TeX book fame. BLUe is its cousin, adopted by me. The idea is that like in Math the basics are explained to jump off from.

² I don't know why ligatures are not handled correctly in tugboat.sty (for example '?', TB 381).

³ These examples can also function as samples.

```
%
%2. To handle numbering and
%   verbatim file inclusion
\everyverbatim={\numvrb}
\thisverbatim={%
  \catcode'\!=12 %is default escape char
  \catcode'\|=12 %is used for inline verb
  \input vrb.tex
  \makeescape\*}%To terminate
\verbatim
Extras after file
*endverbatim
%
%3. To restart (line)numbers
\thisverbatim={\vrb|in0 }%Or \numvrb
\verbatim
Just some text with
ligatures such as '?' switched off (TB 381).
line numbers restarted via
\vrb|in0 (in general via \numvrb).
```

After two blank lines.

```
!endverbatim
%
%4. Inline verbatim via minimal tags
%Minimal | tag for inline verbatim
\makeactive|
\def|{\bgroup\setupverbatim
  \the\everyverbatim
  \the\thisverbatim
  \def|{\egroup\thisverbatim{}}
\thisverbatim{\emc}
Before |<inline verbatim>| after.
```

Basic to the design is the concept of `\this<foo>`, which is analogous to Knuth's `\every<foo>`. With respect to switching the verbatim mode off, I adopted Knuth and Levy's use of the escape character. The mode is finished essentially by `!egroup`. This is a royal road towards simple coding. An escape character is often needed after all, so why not make use of it?

When I looked carefully in `TUGboat.sty`'s verbatim coding I found that the verbatim text is stored as parameter, and therefore restricted. In `manmac` the verbatim text is set within a `\vbox`, and therefore practically also limited. I thought of line-by-line processing but abandoned the idea once I appreciated the idea of

```
verbatim mode
\everypar, and
the use of the escape character
```

for termination.

Furthermore, I like the idea of hooks via `\preverbatim` and `\postverbatim`, to provide the user the opportunity to supply code to be executed before and after. In practice I hardly used the latter possibilities, however.

A bonus which comes with the escape character is that it can also be used to control the counter for the line number in numbered file verbatims. For example the line numbers can be suppressed for parts or appropriately initialized, as demonstrated in the listing of `vrb.tex` at the end of this paper. This approach is very useful when the macro file is appended by its contents with references to the line numbers in order to locate macros.⁴

⁴In studying `Manmac`, the `TUGboat` styles I missed the table of contents with references to the line numbers of the code.

All what has to be done is to include `%!numvrb` in the file when one likes the next line to start numbered, or to include `%!vrb|in100` when one likes to give the line counter the special value 100. To complete it one can suppress the numbering via inclusion of `%!nonum`. This approach does not hinder the original purpose, because it is done via comment lines, and extra comment lines are anyhow wanted to group code parts. Ça va sans dire that this looks nice to me.

4 Conclusion

The macros obey the adage:

prototype, revise and rewrite.

My alternative codings to the verbatims of `TUGboat.sty`—as released in `TUGboat BLUes`—can be seen as the prototypes. `vrb.tex` is an improvement with respect to simplicity, flexibility, speed, independence from the context, and the little use of resources. Confusing was that `\char'174` is the vertical bar in ASCII and `\tt`, but the `\Dash` in `\rm`.

I too found verbatims quite confusing. Now I feel confident, because I understand the process and I only have to remember the specialities of `!` and `|`. To end the verbatim by `<escape char>endverbatim` is also not hard to remember. As usual, things you do yourself look easy. It is only hoped that `BLUe` will feel the same way about it. My case rest.

References

The `TeXbook` and the `LATEXbook` are not explicitly mentioned, because I assume that these are omni-present.

- [1] Knuth D.E, S Levy (1987): The CWEB System of Structured Documentation. (FTP: labrea.stanford.edu, in directory /pub/cweb)
- [2] Laan C.G van der (1994): TUGboat BLUes—how T_EXies do it. MAPS 94.1.
- [3] Salomon D (priv.comm.)
- [4] Schöpf R (1989): A new implementation of the L^AT_EX `verbatim[*]` environments. *TUGboat* 11, no. (2), 284–296.

Appendix: The file `vrb.tex`

Note that `!` and `|` have the meaning of escape character and starting the inline verbatim, respectively. The `!` can be obtained via `!!`, in verbatim mode, and via just `!` outside. The vertical bar can be obtained via `!vrt` in verbatim mode, and via `\vrt` outside. On the other hand when verbatims contain `|`-s, it is easier to change the catcode of `|` into 12 via `\thisverbatim`.

```
%vrb.tex Feb 1994 Version 1.0
%Author: C.G. van der Laan, Hunzeweg 57,
% 9893PB Garnwerd
% The Netherlands
% 05941-1525, cgl@risc1.rug.nl.
%Purpose: Verbatim macros via plain TeX,
```

```

%           to be used with AnyTeX.
%
1 %User toks variables
2 \newtoks\thisverbatim
3 \newtoks\everyverbatim
4 %
5 %User customization
6 \let\preverbatim\medskip
7 \let\postverbatim\medbreak
8 %
101 %User 'options': \numvrb
102 \newcount\vrblin
103 \def\numvrb{\vrblin0
104   \everypar{\advance\vrblin1
105     \llap{\sevenrm\the\vrblin\quad}}}
106 \def\nonum{\everypar={}}
107 %           : \emc
108 \def\makeescape#1{\catcode'#1=0 }
109 \def\makeactive#1{\catcode'#1=13 }
110 {\makeactive\<
111  \gdef\emc{\makeactive\<%
112   \def<##1>{\$ \langle##1 \rangle$}}
113 %
201 %User macro
202 \def\verbatim{\preverbatim\beginngroup
203   \tt\setupverbatim
204   \the\everyverbatim\relax
205   \the\thisverbatim\relax
206   \verbatimgobble}
207 %
208 \def\endverbatim{\endgroup\postverbatim
209   \thisverbatim={}}
210 %
211 \def\setupverbatim{\makeactive\`%
212   \let\!=!\makeescape\!%Knuth&Levy
213   \def\par{\leavevmode\endgraf}%TB381
214   \obeylines\unatcodespecials
215   \obeyspaces}
216 %
217 {\obeyspaces\global\let =\
219 \obeylines\gdef\verbatimgobble#1^M{}}%
219 \makeactive\` \gdef{\relax\lq}}%TB381
220 %
221 \def\unatcodespecials{\def\do##1{%
222   \catcode'#1=12 }\dospecials}
223 %
251 %Minimal | tag for inline verbatim
252 \def\vrt{{\tt\char'\|}}\makeactive\|
253 \def|{\bgroup\tt\setupverbatim
254   \the\everyverbatim\relax
255   \the\thisverbatim\relax
256   \def|{\egroup\thisverbatim{}}}
257 \endinput %14/2/94 cgl@riscl.rug.nl
258 %
Contents
Newtoks
  \thisverbatim.....2
  \everyverbatim.....3
Customing
  \preverbatim.....6
  \postverbatim.....7
Options
  \numvrb.....103
  \nonum.....106
  \makeescape.....108
  \makeactive.....109
  \emc.....111
  <#1>.....112
User macro
  \verbatim.....202
  \endverbatim.....208
  \setupverbatim.....211
  \verbatimgobble.....218
  '.....219
  \unatcodespecials.....221
Inline verbatim
  \vrt.....252
  |.....253
%
History of changes
Febr 1994   Release Version 1.0

```


TUGboat BLUes— how T_EXies do it*

Kees van der Laan

Abstract

The significance of *TUGboat* for the T_EX community at large is praised.

(l)tugboat.sty, tugboat.cmn and the TUG authors' guide are discussed, next to their siblings (l)tugproc.sty and the option eurotex.sty.

Article templates for the various 'styles as is' are provided. Independent and in addition to these I included customing files. Also included is my concrete proposal for a tug.ppt style—for preprints of tugboat.sty, and tugproc.sty, in the spirit of ams.ppt.

Furthermore, a new and simple alternative to the handling of options for T_EX is proposed, based on the toks variable `\this<foo>`, analogous to `\every<foo>`. This is applied to handling verbatims, with as a result a compact suite of verbatim macros to be used with AnyT_EX.

Keywords: Block comment, computer-assisted typography, education, eurotex.sty (option), (L^A)T_EX, macro writing, mark up, optional parameters, preprint style, publisher formats, TTN, tugboat.cmn, (l)tugboat.sty, (l)tugproc.sty, two↔one-column format, verbatims.

1 General

1.1 Introduction

This is the third paper in the series about general, public domain macro collections to mark up and format (typographically) complex math-oriented documents.¹ This delving into literature has all to do with my 'don't rush into code' attitude. I like to study the past and build upon that.

Reality has it that *TUGboat* has functioned as the statehorse of the T_EX-community, and stimulated many a T_EXie. It has also been the example of how a T_EX bulletin should look like for the various LUGs. Whenever the question arose what TUG has to offer its members, the answer has invariably been *TUGboat*. This limited perception has been counteracted since 1990 or so, via the flyer²

'8 great reasons to join TUG'

Moreover, we have now also the portable TTN—T_EX and TUG News. It is perfectly clear what TUG has to offer its members. Real leadership! This paper is aimed at a broad audience

- LUG editorial boards, who like to become informed about the TUG styles
- TUG editors/producers, who welcome feedback
- T_EX hackers, who might learn a lot from the `tugboat.sty` codings and the proposed alternatives
- and, in general those who like to hear what I have to say.

The paper is not aimed at those who just jump into it for search of typos or typographically non-optimal aspects, and who are not really interested in the results of my research.

The TUG styles—and their descendants—represent different points of view. The total of the material treated in this article has therefore the potential of being very confusing, mixing up all those different ideas about macro writing and setup of style files in the mind of the reader. In order to circumvent this to happen I would suggest readers to glance through the introductory material up to the 'tugboat.sty' section, and to skip my ample footnotes and intermezzos on first reading. Then—depending on your T_EX roots—peruse either the parts

- `tugboat.sty`, and `tugproc.sty` with their corresponding templates as provided in the Appendices for plain T_EX trustees, or
- `ltugboat.sty`, `ltugproc.sty`, `eurotex.sty`, and `ttnxnx` with their corresponding templates as provided in the Appendices for L^AT_EX devotees.

And for those who just like to lay hands on the new suite of verbatim macros, jump into 'Alternative: the big deal,' in the verbatim section.³

An inherent weakness of these kinds of papers is that the material treated is dated.⁴ Being aware of this I'll not only concentrate on telling what is in there, but I'll also discuss it, question it, and show alternatives. In short I'll go for

*Parting gift to the BoD of TUG on the occasion of me leaving this board as a special director.

¹The preceding papers in the series are Manmac BLUes and AMS BLUes. BLU stands for Ben Lee User of the T_EXbook fame. BLUe is its cousin which I adopted.

²Contact the TUG office: `tug@tug.org`.

³However, in BLUe's Verbatim I provided a revised suite.

⁴Mind the version numbers of the styles!

providing insight.⁵ When other versions come around it is hoped that the reader can better appreciate these—at least I can now I guess—because he/she knows what is going on, what the pros-and-cons are. In short the lasting value should be that the reader has built a ‘hat-rack’ to appreciate new—or related—versions, the ‘hats of T_EX.’

There seems to hang a sort of taboo—we are the champions, aren’t we?—around these kinds of papers. Invariantly organizations perceive them as an attack on their ‘precious child.’ Let me state it loud and clear: it is not an attack, and definitely not meant as such. If I misperceived anything—unintended :-)) of course—it’s my fault and that has nothing to do with the subject treated or the responsibility of those involved in the design of the style files or TUG<whatever> production.

With respect to the templates I distinguished two issues. First, templates are provided for the various articles, with the use—as *is*—of the <style>.sty-s. Second, there are the customing files, to allow for page sizes, running headers and footers of your own. I rejected the idea of splitting up the article into a bunch of smaller ones dedicated to for example T_EX and L^AT_EX, respectively. In my honest opinion (MHO for short) the total is the most interesting, to have it all together.

To be honest this article is mainly a stimulus for me to

- better understand the TUGboat styles
- give feedback to those involved in the TUGboat et cetera production, and to those who take care of bringing out the proceedings of the various T_EX meetings
- have all the templates and customings together.

This paper is also a parting gift to the Board of Directors of TUG, because of my resignation in June 1994 as NTG’s president I won’t be a special director of the BoD of TUG any longer.

Personally, I consider my new alternative code for the handling of options via `\this<foo>`, which is analogous to `\every<foo>`, very simple and useful.⁶ See the verbatim section in the part ‘tugboat.sty.’

Although this article is critical about ‘How T_EXies do it’ it is tacitly meant as a positive contribution. It can assist the selfinspection process. Be happy with the good news, and be realistic about the *inelegant* ways we mark up, or write T_EX code now and then. Dare to question the a priori choices and coding conventions, adopted a decade or so ago. Re-ponder the effect of wishful thinking especially

in a volunteer-based world. Realize, accept and discuss! This will strengthen our efforts to provide for the best in a multi-dimensional sense.

1.1.1 Warnings!

The paper is incomplete, and certainly not the last word on the issue. It is not easy-reading either. Some 15 years of experience have been glanced through, and the area surveyed is broad. Add to that some new ideas, and that T_EX is inherent complex, subtle, or at least unusual. Moreover, related areas such as literate programming and hypertext, are touched upon. As a result nobody would be really surprised to find that a paper like this is hard to read, when details are also dealt with. At the very least a reader should take his/her time for reading it, and make a selection to start with. It is dated too. Some elements have not been treated equally in all parts, probably because I considered them not that relevant, for the moment. The macro code discussions are different from just inclusion of macro texts, followed by examples of their use. Therefore, I can’t possibly guarantee that they are all correct. However, I have done what I could. I have separately tested the verbatim suite and the templates at the end. Furthermore, my aim has been to convey insight not to provide production alternatives.

1.1.2 Generic coding pitfall

To start with I will discuss a pitfall of mark up coding I tumbled in myself.⁷ In order to have a generic footnote command which automatically keeps track of the footnote counter—to be customized to plain (or L^AT_EX)—I provided

```
\def\ftn#1{\advance\fcnt1\countercurrents
\footnote{$\the\fcnt}$%.typesetting
}{#1}}\newcount\fcnt
```

See what happened with respect to plain’s footnote?

The footnote text is no longer processed on the fly!?!

I should have omitted the argument—*mea culpa*—just a parameterless shortcut. T_EX is so unusual! Deteriorations like these are easily coded and innocently passed on, which is even worse.⁸

1.1.3 Why?

There are a few reasons why it is worthwhile to study TUGboat’s styles. The styles

- embody some fifteen years of experience
- are used in practice
- evolved, driven by the needs of advanced T_EXies
- facilitate submissions marked up in the T_EX or L^AT_EX spirit

⁵Although, to paraphrase Hamming: The purpose of literate programming is not yet in sight.

⁶Comparable in functionality to Knuth&Levy’s verbatim mode.

⁷In good company, though. L^AT_EX’s footnote suffered from the same pitfall. (It is also in `\@makefnstext` in the `ltugboat.sty` section.) I did not stumble on things like this in `tugboat.sty`. It is mentioned at the beginning of this article to illustrate the subtleness of macro writing in T_EX. T_EX is so unusual.

⁸BLU might say ‘Who cares?’ Well, I do and *you should too*, because in the wrong case you can’t mark up verbatims in footnotes as you are used to. Nobody—certainly not flexible and tolerant T_EX—will complain. To be concrete verbatims like `|<special character(s)>|` will go wrong, in the sense of unexpected results. (The use of the alias tag ‘|’ is irrelevant.) A further enhancement is to use a `\global` advance of `\fcnt`. Furthermore, this example is fundamental. It stimulated me to change my `btable` macro into a two-part macro, such that verbatims—read special characters—can be processed too. I also appreciate Knuth’s `\beginchart` and `\endchart`—from `manmac`—much better now.

- allow switching from one- into two-column format and vice versa⁹
- allow inclusion of files in verbatim (This enhances consistency of the macros used and those listed. Obligatory for authors who publish about (L^A)T_EX in (L^A)T_EX.)
- have implemented a general and orthogonal optional parameter mechanism, and allow for short command delimiter forms
- `tugboat.sty` shows how to handle arguments as stored information and more importantly how to process ‘arguments on the fly’
- some macros/utilities can be used in other contexts, for example abbreviations and logos (this is trivial)
- `ltugboat.sty` is an example of how to customize L^AT_EX’s `article.sty`.

1.1.4 What?

First I’ll pay attention to the provided functionalities and answer the question whether it is easy to switch between the styles. Then I’ll discuss how the results in print look like, how the production process goes, what is provided by the style files, what I know of the design, and finally I’ll distill most of the coding conventions. Next the `<style>` files are discussed each as a separate entity. At the end I’ll also discuss *TUGboat*’s descendants

- `(l)tugproc.sty`: the styles used for the proceedings of the TUG annual meetings
- `eurotex.sty` the option used for the EuroT_EX meetings
- and, en-passant touch upon TTN’s style.

Each style has a users’ guide, of which the article of Beeton and Whitney is the most widespread. Barbara Beeton has surveyed the production history in her ‘*TUGboat* production: T_EX, L^AT_EX, and paste-up.’ Copy can be submitted with

T_EX- or L^AT_EX-oriented markup.

In the ‘Appendix: Templates’ I provided a sequence of template articles—the `<style>.tem` files—which show the essential markups. This anthology makes it easy to compare the various markups. One can’t get around the inconsistencies, alas. In the ‘Appendix: Customing’ I provided the `<style>.cus` files, for modifying the page size and the running headers and footers. The ‘Appendix `tug.ppt`’ contains my proposal for a preprint style for TUG (and LUGs). The last appendix contains the table of contents.

Functionalities. In a publication I need for markup

- at the outer level: title part, keywords, abstract, headings, floats, footnotes, margin notes, appendices, bibliography, and index
- at the inner level: fonts (coupling and selection), special paragraphs (quote, lists), (file) verbatim listings, programs, pictures (graphics), tables, display math, automatic numbering, (symbolic) cross-referencing, and citations.

Functionalities: (l)tugboat.sty

	<code>tugboat.sty</code>	<code>ltugboat.sty</code>
title part	++	++
keywords	plain	+
abstract	plain	+
headings	++	++
floats	plain	++
footnotes	+	+
margin notes	discouraged	L ^A T _E X
appendices	plain	+
bibliography	plain	+
.....
font	plain	NFSS
quote	+	+
lists	++	+
verbatim	++	++
file verbatim	++	+
escape verbatim	++	+
programs	none/plain	none
pictures	none/plain	+
tables	plain	L ^A T _E X
display math	plain	L ^A T _E X
aut. numbering	none	L ^A T _E X
(sym) cross refs	none	L ^A T _E X
col switching	in progress	L ^A T _E X
citations	none	L ^A T _E X

++ excellent + good

This thinking at two levels is fundamental, although the borderline is ill-defined. In the accompanying table I have enumerated these functionalities and indicated roughly to what extent they are supported by `tugboat.sty`, respectively `ltugboat.sty`.

From the table it appears that the question

Whether switching from `tugboat.sty` into `ltugboat.sty` or vice versa is easy?

is ill-posed, because the styles support different functionalities, of yet.¹⁰

The prime developers are Ron Whitney and Barbara Beeton for `tugboat.sty/cmn`, and Adrian Clark, Frank Mittelbach, and Rainer Schöpf, for `ltugboat.sty`, all coordinated by Barbara Beeton, the editor. The TUG styles

⁹Not provided in full generality of yet—in test phase though—that is, it is not possible to switch within an arbitrary page. `tugboat.sty` allows one-column at the first page, and L^AT_EX—and thus `ltugboat.sty`—clears the page after a switch. Two-columns is not supported by the `amsppt.sty` style. In `manmac` it is used only for formatting the index (no floats in there!). For L^AT_EX Frank Mittelbach has released `multicol.sty`, and I know of some adaptations for plain.

¹⁰Nevertheless, I make it a habit to mark up headings via `\head*...*` et cetera and to precede in-line verbatims by `\verb`. (The latter when working with T_EX redefined as empty.) Similarly, I use `\small` and `\tiny`, next to `\thepage`. The footnotes are marked up by `\ftn` to be customized to plain or L^AT_EX, given the context. For the markup of control commands I use `\cs`. A first step towards generic—context-independent—markup.

are owned by the T_EX Users Group, and released in the public domain.

1.1.5 Notations and definitions.

`\ea`, `\nx`, `\aa` and `\ag`, are used as shortcut notations for `\expandafter`, `\noexpand`, `\afterassignment`,¹¹ and `\aftergroup`, respectively. TUG denotes the T_EX Users Group, and LUG stands for Language or local Users Group. Text borrowed from the users' guides and the style files is surrounded by single quotes. (I never use double quotes.)

!?! denotes that I'm much surprised. ?? denotes that I can't believe my eyes.

Orthogonal facilities mean independent facilities which can be used as such, but when combined will yield new possibilities, without extra coding.

With respect to the use of fonts I'm a bit loose. When I denote *TUGboat* it might mean the journal, the editors, the readership, . . . , depending on the context.

1.2 What does TUGboat look like?

First of all it looks great! More seriously *TUGboat* is two-column biased. With respect to the contents major parts are: general delivery, philology, fonts, graphics, book reviews, typesetting on PCs, tutorials, letters, macros, abstracts, news and announcements, late breaking news, TUG business, forms, advertisements. This list reflects the outer level of the table of contents, as supplied on the back. (This classification is distilled from a *TUGboat* issue, and it might be that there is no rigid classification decided upon.) The following issues are also distilled from the copies I own.

Major parts start with the title framed. Each contribution within a part is separated with a `\hrule` from the previous one.

From the outer level point of view each article has the usual structure

- title part (title and author name)
- copy proper (abstract, sections, acknowledgements)
- back matter (references, appendices, signature (name and address)).

There is no uniformity with respect to section numbering and punctuation of the titles.¹² The ordering of the items vary in the back part, for example, the signature is not al-

¹¹ Yes I know, the control sequence is already in use for the Ångström unit, or to denote the å character of the Scandinavian alphabet. I don't use those in here. Just a mnemonics shortcut that is all.

¹² The use of `\nopunctuation` suppresses the end dot after a subhead title. Then none of the heads gets automatically an end dot, not even the running-in subheads. This is simple to remember. From a functionality point of view there should be no default dot. It must be left to the authors whether they need, for example, questions (with question marks) as titles of a heading.

¹³ At the end of the users' guide it is recommended to provide for abstracts. (Abstracts are provided in German at the end of each *TUGboat* issue, from 1990 or so onward.) It is much clearer to have it included in the 'outer form' list and incorporated in the template to come. Barbara Beeton communicated that one of her wishlist items is to provide for Author Guides, similar to those of AMS. Volunteers go for it!

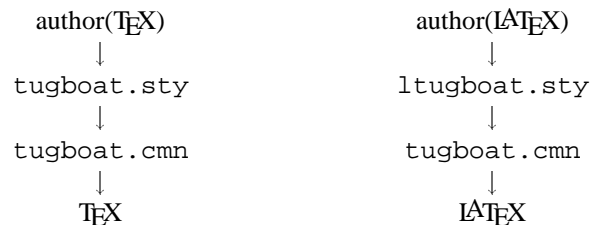
¹⁴ Note that this is an oversimplification, it restricts attention to T_EX and L^AT_EX, and abstracts from METAFONT interaction, the inclusion of PostScript, and the like. As I have heard on many an occasion 'TUGboat production is not simple.' There is nothing against it to try for making it simpler, isn't? In MHO, and with all respect, there is no other way.

ways at the end. Not every article has an abstract¹³ or a keywords part. References are set differently by authors. Labeling and citations vary. As floating objects (insertions) we have page-wide tables which appear at the top of a page or at the end of the article, next to small one-column tables. Footnotes are column-wide.

Remark. Of course there should be 'exceptions to the rule,' especially when an article is about a special formatting issue and the result is demonstrated by the article itself. As an example one could think of Mittelbach's multi-column article. Also some authors have insisted upon one-column format throughout.

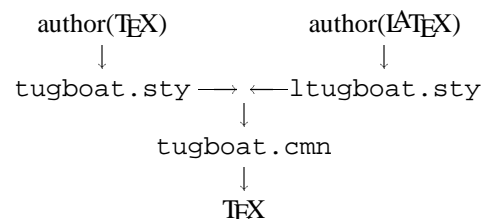
1.3 TUGboat processing

The T_EXnical production of *TUGboat* can be summarized¹⁴ by the following streams



Next to this there is a refereeing system (from 1990 or so onward). Multiplication of the blues is done by a printing house. The kernel of the editorial board has been proven to be stable and has been in charge for many years.

An alternative approach is there a reasonable one? Let us not bother about the practicalities and concentrate first on how it could have been done best, T_EXnically. An alternative is to allow for a L^AT_EX author interface next to the T_EX one—with common lower level markup—as depicted by the following scheme



Superficially this looks the same but it is not, because there is no L^AT_EX in there. The *TUGboat* look-and-feel in print

is inherently guaranteed, and the \LaTeX user can use the commands he is familiar with.¹⁵

However, this idea is not viable because the styles differ significantly by the provided functionalities. Moreover, it is no sinecure to reimplement the \LaTeX features in `tugboat.sty`, or the other way round. Furthermore, the temptation will invariably be to do it in the best way thinkable, and that for sure will take time, as we can witness from the $\text{\LaTeX}3$ project. However, some if not all, has been done by Spivak in his $\text{\LaTeX}S\text{-}\text{\TeX}$. Hmmm, puzzling.

Another idea is to start from a generic style which can be coupled to `tugboat.sty`, \LaTeX , or you name it. The latter approach is interesting for authors because *their* interface remains the same.¹⁶ De facto \LaTeX functions more or less this way. That is

$\text{\LaTeX} \equiv$ the *de facto* user interface.

The point I like to get across is that it would be nice when \TeX and \LaTeX marked up submissions can be processed with the ‘markup in the small’ by \TeX . For example for math and tables the macros from plain can be used.¹⁷ \LaTeX ’s variants for these are syntactic sugar, in MHO with all respect. No extra facilities have been added there. \LaTeX ’s greatest contribution is the users’ guide aspect in contrast with the \TeX book which contains it all. Lamport ‘dared to do less.’

An annoyance is to have to remember that the markup of a reference to a *TUGboat* article needs awareness of whether I’m using `tugboat.sty` or `ltugboat.sty`. This because of the following borrowed from `tugboat.cmn`.

```
\ifx\tugstyloaded@plaintextstyle
  \def\tubissues#1(#2){\TUB~#1, no.~#2}
\else
  \def\tubissues#1#2{\TUB~#1, no.~#2}
\fi
```

Why? There is no reason for ‘upward compatibility.’ It sins against the generic idea. In MHO—with all respect—these kinds of low level

markup should and *must* be independent from the context.

There are more such context dependencies, as mentioned in an earlier footnote. Other discrepancies come from `tugboat.sty`’s `\article` versus \LaTeX ’s `\maketitle`. Why not redefine `\@maketitle` and supply this redefinition in `tugboat.cmn`?

¹⁵So the question arises whether it is feasible to start from a set of specs (very close to `tugboat.sty`, I presume), ‘implement’ these as `tugboat.sty`, and provide a \LaTeX user interface on top. From the maintenance point of view the answer should be yes. From the human factor point of view the answer is apparently no. The financial means are modest too. I pity that. For a finger exercise of the approach see the section ‘Lists,’ in the `tugboat.sty` part, for variants with respect to \LaTeX ’s `\itemize`, based on Knuth’s `\item`.

¹⁶A disadvantage is, however, that the appearance will vary with the coupled style. When we look at it from the preprint point of view then this disadvantage turns into an advantage. We want then different appearances in print!

¹⁷Let us assume an appropriate authors’ guide exists.

¹⁸On second thoughts one should not enforce at all, always leave the choice with the user.

¹⁹Note that I did not say anything about *what* is stated in a referee report. Too many opinions on that issue, and that is the referee’s freedom. But first the form things, the trivia. And if that is settled we can go for the quality aspects of the contents. When the form aspects are not sufficiently settled it is of no use going further.

²⁰A lot of thought has been given to consistent coding of the various tags! Top-class, but perhaps too smart.

Agreed, for most authors it does not hinder. They follow one of the streams.

My rule for issues like these is: When in doubt to enforce one way or the other on your user, *don’t!*¹⁸

Refereeing is in MHO—with all respect—a bit misplaced and takes too much energy. Roughly the same results can be obtained when submissions are accompanied by a recommendation of a friendly colleague. This approach is more in agreement with the character of a user group: to assist each other, instead of rejecting—rightly or wrongly—a contribution. And this is much more efficient too. Furthermore, who referees the referees? And what for objective selection process is there to become one? I blame the current process from experience, where despite the guidelines for referees the referee reports vary enormously, not in the least in structure, items treated and tone.¹⁹ I guess the guidelines are insufficient and should pay more attention to the right, respectful and helpful *attitude* of the referees. It remains a difficult and delicate matter. I know of the wishful arguments that the journal status is sought, to benefit from belonging to the class of ‘refereed journals.’ Perhaps, I’m a bit naive but the quality of a journal is judged by the peers in a field. In the (small) area of a specific computer language—and \TeX can be seen as such a language, but admitted wholeheartedly a brilliant one—I have not heard of a ‘real journal,’ and don’t believe *TUGboat* will become one, simply because we need a broader umbrella, for example a journal about electronic publishing like EP-ODD. Don’t suboptimize!

It may look like that I’m treated badly—there is some truth in there ;-)—but that is not the point. What I like to get across is that it has all to do with

get your priorities right and delegate

especially when there is still so much to do for ‘finishing’ and polishing the styles. Furthermore, it is a volunteer-base world, meaning we are always short of volunteers, and invariably suffer from lack of time.

1.4 Design

The *TUGboat* styles have not been ‘designed.’ That is: to my knowledge there is no description of how the journal should look like in its total, the specs so to say to which the style files had to conform. It started with ‘a package based only on plain,’ and²⁰

‘...later, as demand for style files follows wherever L^AT_EX-devotees wander...’

a TUGboat variant of L^AT_EX’s `article.sty` was also created. This has been called `ltugboat.sty`. Common matters have been split off into `tugboat.cmn`.

The users’ guide mentions euphemistically²¹

‘The two macro sets yield much the same output, differing in certain ways for input.’

No results have emerged in public so far with respect to the markup of bibliographies, pictures, symbolic referencing or citations other than L^AT_EX’s approach.²² The recent practice of handling encapsulated PostScript has not been included in the users’ guide of yet. The latter is used in practice by some authors already, as was communicated to me by Barbara Beeton.

To end up this section I still believe that specifications are mandatory,

and embody ‘half of the work,’ if not more. Furthermore, I would write nowadays style files in the form of literate programs, as suggested by the polish GUST at Aston, based on specs. Furthermore, by this approach as communicated by Włodek Bzyl the descendants can be implemented via change files. That is all. Neat isn’t it?

1.5 T_EX coding conventions

For `tugboat.sty` a strict uniform and orthogonal T_EX coding philosophy has been adhered too. For `ltugboat.sty` L^AT_EX’s `article.sty` has been used and ipso facto its coding style. In view of the L^AT_EX3 project we might expect a big change with respect to the latter.

1.6 What is provided by the styles?

From the authors’ guide for TUGboat—which by the way is a nice paper in itself—we see that the following functionalities have been provided for

- at the outer level: title, author, address, article proper (section structuring with paragraphs (quotes), lists, verbatims, floating figures), and signature
- at the inner level: the details of display math, tables and figures are not described. One is referred to T_EX (respectively L^AT_EX) or specific macros.

Tags for keywords, abstract and bibliography were introduced along with L^AT_EX.

1.6.1 Contents of the style files

`tugboat.cmn` contains among others the formatting of abbreviations like `\AMS` for American Mathematical Society, `\aw` and so on. Very handy for enhancing consistency.²³ Furthermore, it contains macros et cet-

era common to (1) `tugboat.sty`. The list of contents reads

- helpful shortcuts
- abbreviations and logos
- utility registers and definitions
- section heads
- registration marks
- miscellaneous useful stuff
- dates and other items which identify an issue
- issue number in file name
- authors, addresses, signatures
- hyphenation examples
- contents
- change history.

Some 900 lines.

`tugboat.sty` contains

- fonts
- page dimensions
- headers/footers
- page adjustments
- output
- general mechanisms for tags
- titles, authors, addresses
- heads
- text and subtext
- lists
- verbatim
- figures
- utilities
- initialization
- history of changes.

Some 2300 lines.

`ltugboat.sty` contains

- prevent double loading
- fonts (accounts for plain L^AT_EX and NFSS)
- normal tugboat dimensions
- lists
- titles, authors, addresses, signatures
- Customing of headings to look more like tugboat
- footnotes
- figures
- quotes
- bibliography
- running heads
- OTR
- selfdocumenting style
- miscellaneous defs for compatibility with `tugboat.sty`
- change history.

Some 600 lines.

²¹ A bit of wishful thinking—or planning ahead, or whatever you like to call it—for the moment because the functionalities provided differ much. In `ltugboat.sty` it is stated for example ‘Redefine style of section headings to look more like TUGboat. Start with definitions from `art10.sty`. (Only `\section` so far.)’ See also the table ‘Functionalities: (1) `tugboat.sty`’.

²² Sigh, not enough time and/or volunteers. The wish list is unknown to me. If known it would help possibly, apart from the difficulties to adhere to the TUG coding discipline.

²³ This list of abbreviations and logos—with the coded formatting conventions—can better be provided as a separate file to be used by the T_EX-community at large, also with formats different from TUGboat.

As can be seen the latter two files are not similar. From a logical point of view more can be split into the `cmn` file, especially fonts coupling and headers and footers handling.

I consider the selfdocumenting style as a hobby horse, out of context.

1.6.2 One-ness

Now and then the wish pops up to include all the style files in one system, to encapsulate codes and documentation, better known as literate programming nowadays. Włodek Bzyl and Tomasz Przechlewski just did so by transforming it into WEB. (See their TUG '93 article.) The bonus which comes with this approach is that one can easily include change files—they will also become part of the set, and the maintenance annoyance is reduced. An index is by default available, which makes the looking up of commands easier, in principle. At the moment it is unclear to me, within the WEB approach, how for example the 'abbreviations' module can be made separately available, to be used in other contexts without the overhead of the total system. It is even darker for me how to make the verbatim functionality separately available. With respect to the customizing I agree that the change file concept is beneficial and worthwhile, and in general I believe in the approach. Barbara Beeton communicated that work is in progress to provide it all with `doc-option`, Mittelbach's style option to format documentation of style files. Puzzling is still why not as a WEB, as suggested by the polish GUST at Aston last year.

1.6.3 For editors only

It would be nice to organize the style files such that there is a part marked 'for editors only.' And why not indicate in the same spirit parts 'to be removed/replaced in version such and such,' and the like?

1.7 Conclusion

A first-order conclusion is that `tugboat.sty` and `ltugboat.sty` provide superficially the same appearance in print, but differ in the markup and the macros used: `tugboat.sty` respectively \LaTeX .

For the communications of a user group the refereeing of TUGboat submissions is misplaced.

Provide authors not only with the good users' guide but also with a template with the obliged items in the right order. And what about a simple preprint style?

Provide a list of abbreviations (logos) as a separate file to be used with any format or style.

With respect to developments my wishlist for `tugboat.sty` is

- provide specifications, as a blue print for development
- reshape the style files into a CWEB with documentation and index encapsulated

- provide a preprint style (for a proposal see Appendix C)
- allow for customings via change files
- general two \leftrightarrow one-column switching
- generic *simple* bibliography tools (like my BLUE's Bibliography), better still maintain the pre-formatted database at the editor's site, with authors only providing entries to this database
- to make *simple* verbatim functionalities separately available for use with Any \TeX , especially \LaTeX
- provide automatic numbering schemes
- symbolic cross-referencing²⁴
- common fonts coupling
- keywords and abstract environments
- provide emphasize tools for in-line, for example `+. . . +` with `\eminline. . . \endemphasize`, and in display, for example via the use of `\preemphasize` and `\postemphasize`
- provide a picture environment, for example start from the macros as supplied in `gkpmac`
- document the use of encapsulated PostScript.

With respect to `ltugboat.sty` my wish is to have that more compatible with `tugboat.sty`, that is allowing a \LaTeX user interface.

1.8 What does the TUG Annual Meeting proceedings look like?

Great, especially the last one—1993—set in Lucida. A main difference with the regular TUGboat is that the titles with *complete* author(s) information span two columns (left justified), along with the (centered) abstract. (Keywords are tacitly discouraged as I experienced. Mine were commented out!?) The appendices are in one-column and start on a new page. Page numbers and conference information are set at the bottom of the page. (`\midpage` has been recoded in order to allow for running feet.) Footnotes are set per column as in `tugboat.sty`. The headers and footers are different—contain other information—from the regular TUGboat issues.

1.9 Processing proceedings TUG Annual Meetings

Basically the same procedures are followed, but handled each time by a different team of proceedings editors. As can be expected there is not much room for experience to be built up, with all respect to the editors who have done a tremendous job in the past.

Refereeing is done too. For the TUG '93 Annual meeting I was surprised to experience that the papers were first refereed and after being accepted there was a second refereeing process—with different referees—in order to warrant the quality for the submission for the proceedings?!? Perhaps the confusion arose because I had the papers already finished when only abstracts were requested. I should have known better, not to submit at that time the paper

²⁴I know of Spivak's— \LaTeX —rich, independent and general referencing scheme. Włodek Bzyl communicated that the macros used for formatting 'Concrete Mathematics'—`gkpmac`—also embody such a macro tool, which can be used along with Any \TeX , and therefore `tugboat.sty`. There, however, `|` is used and we can't use them as such together with `tugboat.sty`.

too. However, there is much miscommunication here. For EuroTeX-s I had to judge ‘contributions’ submitted as abstracts, complete papers, and even non-submissions?!? The point I like to make is that there is much confusion around and when an author expresses his misunderstandings accept that as *a reality*, whatever the intentions or setup. Accept and say, OK how can we cope with this, it has not been intended, instead of rejecting and saying it is not true.

The **proceedings styles** are less stringent coded (with all respect). Especially, those which build upon `tugboat.sty`. The latter don’t follow the conventions—in full generality—of the tags. Apparently these design conventions have been considered as overdone. The needed functionality has been provided for in a simpler way, superficially similar. See the ‘`tugproc.sty`’ part.

`tugboat.sty`

2 TUGBOAT.STY

Given the users’ guide the use of the commands (see Beeton and Whitney, p.379 for the outer form commands, and p.383 for a command list summary) won’t give much rise to difficulties in using them. Good work! The version discussed is 1.14, Feb 93.²⁵

The `tugboat.sty` is strong with respect to handling of (semi-transparent) verbatims especially the file verbatim (reading and writing), the possibility to use short command forms, the handling of options, and the possibility to delimit the argument by `*-s` or just the end delimiter `\end<foo>`. The optional command handling does not check. It functions like a mechanism for including user guidance.²⁶ This style should be strong—and undoubtedly it will be in near future—with respect to switching from two-column into one-column format and vice versa.²⁷

2.1 Customing

2.1.1 Page size

The page size parameters involved are summarized below (restricted to one and two column)

```
\normalcollgt=54pc \collgt\normalcollgt
\pagewd=39pc
\twocolwd=18.75pc \intercolwd=1.5pc
```

²⁵ Still current of Februari 1994.

²⁶ Ron Whitney communicated that the idea behind this not checking of the user supplied option against the allowable options, was to allow freedom for the user, to include whatever he/she wishes only restricted by that it shall be done via the option gate.

²⁷ The versions I studied did not contain it in full glory, yet. In MHO a two-column format must allow for tables or pictures which are page-wide, entailing switching to one-column and back. As said elsewhere I like Knuth’s approach: one-column the format to work with and now and then provide for pseudo two-column format as implemented in his `manmac` macros, as demonstrated by the `TeXbook` itself. This is simpler and more efficient. A matter of taste? (GUTenberg cahiers switched from the two-column style into one-column style last year or so. NTG’s MAPS is two-column biased. The EuroTeX ’88, ’91 and ’92 proceedings adopted one-column.) Definitely context dependent. Some people have strong opinions about the optimal width of lines of text. The format for scientific books is different from the format used for journals (and newspapers). The latter prefer multi-column format.

²⁸ It is a ‘white lie,’ only part of the truth. We also need ‘markup in the small’ for example for math, tables and graphics.

²⁹ Don’t look for the answers—if any they are implicit—meaning you have to judge yourself.

2.1.2 Headers and footers

Headers and footers are set by invoking `\rtitle` and `\rfoot`, respectively, with as replacement text `\hbox to\pagewd{...}`.

2.2 Coding, or TeXies at work

I can’t agree more with²⁸

‘... The tags whose use we encourage are the higher level tags that mark the logical document structure. ...’

However, there are a lot of questions: whether the right a priori choices have been made, whether the goal has been attained and whether the coding used is the best? Is it easy to adapt, to extend? Can we learn from it? Should it be followed? In order to answer these questions let us consider some elements in detail.²⁹

2.2.1 Fonts

Plain’s handling of fonts has been adopted. From the file (abridged)

... The fonts—5-10pt, roman, slanted, bold, teletype, extension (math)—are sufficient for most ordinary *TUGboat* productions. Additional titling fonts are defined elsewhere, and occasionally an extra font will be needed for a particular item (e.g. device charts) and defined in that file. ...

With respect to the recent NFSS the following

‘... Anticipated changes to this font handling scheme: Dynamic loading of fonts, probably in groups according to size. ...’

From the TeX coding point of view the adding to a `toks` variable is interesting. It is based on the `toks` extension—`\t=\ea{\the\t *}`—as given on p.373–374 of the `TeXbook`.

```
\def\addto#1#2{\csname @addsto\string
#1\endcsname=
\ea{\the\csname @addsto\string
#1\endcsname#2}}
%with
\def\@additionsto#1{\ea\the\csname
@addsto\string#1\endcsname}
```

With, for example, extension via

```
\addto\tenpoint{\def\ssf{\tenss}}
%and use via
\@additionsto\tenpoint
```

Quite something to have a backslash as part of the name of a `toks` variable!

2.2.2 Conventions

tugboat.sty's systematics manifests itself in the general mechanisms for

- the markup via begin tags with the short delimiters—*s—or the end delimiter \end<foo>
- keeping track of the environments via \CurrentTag³⁰
- the systematic invocation of the execute macro—logically \@begin<foo>—via

```
\csname@begin\CurrentTag\endcsname
```

- the provision of defaults and how to override them, for example

```
\@itemtag={\bullet$}
\def\tag#1{\@itemtag={#1}}
%with use
\tag{<your itemtag>}
```

- eliminating superfluous spaces and blank lines from input, for example via the use of³¹

```
\def\DeleteOptionalSpacesandPars#1{
  \@ignoreall{ }\@ignoreall{\par}{#1}}
```

- handling of optional arguments (attributes)
- the inclusion of \every... , for example, the powerful \everyverbatim{\enablemetacode}, to handle metacode in verbatims throughout
- general attributes (for example \ruled, \numbered, to be orthogonally used with various constructs)
- the handling of the separation with what follows after an element via \@next
- the two ways of handling an 'argument': via storing in \@argument and use it later, or on the fly (the latter—characterized by \@savingargumentfalse—has the advantage that the catcodes have not been set).

Great!

Outer form tags are implemented in a *consistent* and orthogonal way. All the tags obey the same syntax. After the begin tag optional parameters can be used independent of whether the short-form delimiters or the end delimiter is used. Quite a T_EX coding marvel! However, the negative side is the too rigid discipline.³² Even for simple things like \head*...*—where at the user level there is no much need for optional parameters—the rigid discipline has been adhered too.

In this case, however, all what has essentially to be done is typesetting of the head title in an appropriate typeface with sufficient white space around it, and paying attention to the right penalties in order not to let the head title start at the bottom of a page.

So, this beautiful discipline has been applied at the expense of simplicity and flexibility now and then. Which is quite

understandable. Furthermore, I for one don't consider the coding of the title part relevant. Of course, the creators of the styles have to worry about it, but I won't. Everybody rides his hobby here. I use templates for these parts given by the various sources, whatever the codings.

2.2.3 General mechanism for tags

An author can choose from the following variants to mark up his copy with respect to the tag \<foo> via

```
\<foo>* <argument>*
\<foo>[<options>]* <argument>*
\<foo><argument>\end<foo>
\<foo>[<options>]<argument>\end<foo>
```

with spaces absorbed where necessary, especially around *s. From the file tugboat.sty the following background to the process flow has been copied.

Upon sensing an opening tag (call it \<foo> here), the following process is set in motion.

1. \beginngroup (so definitions and settings are localized).
2. The default situation for \<foo> is set up.
3. If appropriate \every<foo> list is read (this allows one to override the tugboat.sty factory defaults).
4. Optional commands are read, the '\', '{', and '}' are restored to their status appropriate to \<foo>.
5. The \@begin<foo> macro is executed. This may involve branching dependent upon flags set by options. It may also be a place where spaces and carriage-returns are activated.
6. The 'argument' to \<foo> is read and stored or processed on the fly (the method employed is generally fixed for each tag). The argument may be delimited by *...* (called the 'short-form' here), or supplied up to ... \end<foo> (called the 'long-form').³³
7. A cleanup macro is executed which also ends the current group. This may do all the work if an argument has been read and stored.'

From the users' guide the following operational scheme has been copied.

```
<read tag>
\beginngroup
\set defaults>
\the\every...
<read options>
<branch to appropriate action,
  using 'argument' as necessary>
<cleanup>
\endgroup
```

From one point of view the uniform coding scheme is attractive: once understood, one can read the T_EX coding of all the tags. Hmmm, is that true? For me it is not true, because it is too difficult to really understand the coding. It costs too much time or energy to master, while

³⁰This is used for the invocations of the specific \every<foo>, \@begin<foo>, and \@end<foo> commands. With elaboration via the processing on the fly branch there is no check on the correspondence of the \begin<foo> and \end<foo> tags.

³¹This looks strange especially in view of plain's \ignorespaces, and that spaces are deleted after control sequences anyway. Ron Whitney communicated that what comes after is executed, and that is needed now and then. Hmmm?!?

³²Apparently too difficult for epigons, as demonstrated in customizing tugboat.sty into tugproc.sty.

³³Quite something to do this without the parameter separator mechanism! Moreover, this handling via parsing instead of via the parameter separator T_EXnique is more powerful, because it can gobble optional spaces and pars.

the alternatives are simpler, albeit less uniform. Relevant questions are: Should I really work it through? Understand it thoroughly? Communicate about it? A little pondering prompts me that there is hardly an audience for it except me, despite the smart coding.³⁴

From another point of view the advanced T_EXnology hinders the extension of `tugboat.sty` once the developers are out of touch.

I will try to stay at the outer level and discuss the global flow of what is going on and distill as many T_EX programming paradigms as possible.

2.2.4 Checking ahead

This mechanism is amply used in the `tugboat.sty`. The comments in the style file read

‘... Often we check ahead to determine the next course of action. `\@checknexttoken` is used to check for optional commands, to check for the short-form argument-delimiter, and to ignore characters in certain situations. The macro is just a check; applications must do whatever is appropriate with the ensuing tokens.

`\@checknexttoken`—with three arguments—checks the next token against argument #1. If the two are the same, #2 is executed, otherwise #3. The comparison is done with `\ifx`. Since we check ahead with `\futurelet`, the first argument is stored with `\let` as well. To include the case where #1 may be a space we have to go through a small contortion to `\let\@basetoken= that space. ...`’

`\@checknexttoken` is invoked *directly* in

- `\def^M`
- `\@afterbegintag`
- `\@checkoptions`
- `\@ignoreall, \@ignoreone`
- `\@firstverbchar`
- `\page`.

These kinds of macros are important in T_EX coding because the course of action is determined by the tokens in the input stream, read by the markup. Spaces are unintended now and then so we like to get rid of them. In order to understand the macro the following from the syntax

```
\let<control sequence><equals>
  <one optional space><token>
\futurelet<control sequence><token><token>
%and from tugboat.sty
\long\def\@checknexttoken #1#2#3{%
  \futurelet\@basetoken\iffalse#1\fi
  \long\def\@next{%
    \ifx\@baseis@next
      \long\def\@@next{#2}%
    \else\long\def\@@next{#3}\fi
    \@next}%
  \futurelet\@nexttok\@next}
```

Illustration of function `\@checknexttoken`

Toks after	Match?
abc a	no
xyzx	yes
{ }bc b	yes
{ }bc{ }bc	no!

The results in lines 3&4 in the table look strange but come from the nature of scanning by tokens. The simplified (educational version of the) macro reads

```
\def\chneto#1#2#3{\let\@bt= #1
  \def\@next{\ifx\@bt\@nexttok
    \def\@@next{#2}%
  \else\def\@@next{#3}%
  \fi\@next}%
\futurelet\@nexttok\@next}
```

Explanation. The macro adheres to the usual structure for looking ahead: a `\futurelet\@nexttok\@next` at the end with the process macro `\@next` defined in the replacement text. The first difficulty is the way the first argument is compared to the `\@nexttok`. I simplified the assignment into the use of `\let` instead of `\futurelet`, with the unnatural kludge `\iffalse#1\fi` removed. Second, the documentation says there is a problem in scanning when an outer control sequence is hit. The production macro solves this by introducing an extra level via `\def\ifx@bt@nxt{\ifx\@bt\@nxt}`. A T_EXing paradigm!

2.2.5 DeleteOptionalSpacesandPars

From the file I borrowed

```
%Execute #1 after ignoring spaces
\def\DeleteOptionalSpaces#1{%
  \@ignoreall{ }{#1}}
%Execute #1 after ignoring spaces and \pars
\def\DeleteOptionalSpacesandPars#1{%
  \@ignoreall{ }{\ignoreall{\par}{#1}}}
```

On first sight it seems that there are superfluous curly braces. However, the definition takes two arguments. Apparently there is a nested invocation. Pondering about the macros reveals that there is an ‘error-correcting’ process inserted too. The parameter #1 is known at design time. So, whatever follows, this #1 is executed in reality. Hmmm, is this a really beneficial side-effect? Barbara Beeton communicated that ‘... this macro permits for a blank line to be left after every section heading of whatever level without having to worry whether it is ‘paragraphed’ or run in. This is provided almost entirely for the sake of a simple and robust user interface, one that is forgiving of reasonable variations in user practices. ...’

The following is a comment from Ron Whitney which shows the subtleness involved.

‘... The `\@ignoreall{ }` functions somewhat differently than `\ignorespaces` in that it gobbles all spaces and then executes a given command (as in `\@ignoreall{ }\<foo>`). If `\<foo>` itself takes arguments, one wouldn’t want

³⁴When I finished working on this paper, I realized that there is definitely a need for these kinds of discussions and clarifications. New TUG annual meeting proceedings editors should peruse the paper and also study the `tugboat.sty`. An editor should understand that style as a prerequisite, how about that? (No new editors anymore?)

“\<foo>\ignorespaces” in place of this. Perhaps you’re—me thus—thinking that \<foo> will ignore spaces anyway in searching for an argument, but this depends upon how \<foo> is defined, and I felt the other approach was uniform and provided something stable from which to work...’

Pondering about the desired functionality I can imagine that the problem arose in implementing the scanning of a stream of tokens with in certain contexts a loose syntax with respect to spaces and blank lines, for example around *-s. I have nevertheless the feeling that \ignorespaces would do—spaces between \<foo> and argument?!?—especially when augmented with a more rigid syntax in order to get and maintain as simple code as possible. This despite respecting ‘worn-in’ user habits, for which we should always have an open eye and ear. And, after all an author should always *mind the spaces* with T_EX markup.

Whatever your opinion and for the fun of it I have a few alternative codings

```
%1: skip until #1 and execute #1
\long\def\deleteuptoandexecute#1{\long
  \def\gobble##1#1{##1}\gobble}
%2: just gobble #1 repeatedly
\long\def\gobbleall#1{\long\def\fifo##1{%
  \ifx##1#1\else\long\gdef\nxt{##1}\ofif
  \fi\fifo}%
  \fifo}
\def\ofif#1\fifo{\fi\nxt}
```

Remark. The above is straight and works for copy *without outer defs*. The latter can’t be used with fifo with an argument. As Spivak found out earlier outer defs give a lot of ‘pain,’ read complicated codings. Below—for hackers only—I have enclosed a teaser which allows outer defs to be scanned. It has been tested to delete spaces and pars. My first significant application of \afterassignment.³⁵

```
\let\aa\afterassignment%local shortcut
\long\def\gobbleall#1{\let\nxt\relax
  \let\arg= #1
  \def\fifo{\ifx\else\ofif\fi
    \aa\fifo\let\nxt}
  \aa\fifo\let\nxt}
\def\ifx{\ifx\arg\nxt}
\def\ofif#1\let{\fi}
```

A parameterless version is

```
\def\gobbleall{\let\nxt\relax
  \def\fifo{\def\fifo{\ifx\else\ofif\fi
    \aa\fifo\let\nxt}%
    \aa\fifo\let\arg}
  \def\ofif#1\let{\fi}
%with test
marknew\gobbleall\newtoks\newtoks
\newtoks after new
```

³⁵For a production version don’t use the shortcut.

³⁶As follows from below this command does much and much more than just inserting the options. It also executes to the end!

And what about ‘\futurelet \@basetoken \iffalse #1 \fi’? Perhaps naive, but I think the above is equivalent to \let\@basetoken=#1. However, it is ingenious in that it removes #1 from the input stream after the use of \futurelet.

Salomon in his courseware uses basically the same approach with simpler coding, though. He just inserts the options and performs the catcode change of the backslash after that.

³⁷Mind the catcodes! Agreed this is pin-pointing to details, but as my teacher T.J Dekker would say: details matter.

³⁸For the exceptional case that the verbatim text starts with a [, then \lastoption can be used which sets \@lastoptiontrue.

³⁹A consequence is that it is not straightforward to search the file for the use of a specific \@begin{foo}.

A nice addition to my list of applications of the FIFO paradigm.

2.2.6 Options

The options are parsed via |checkoptions|,³⁶ which at some stage invokes \@readoptions[#1]{#1...}.

Note that the replacement starts with #1, so ‘the option’ is just inserted,³⁷ and there is no checking!

Aha, if so there is a much simpler T_EXnique for this optional parameter functionality. Namely, in analogy with

```
\every<foo> introduce \this<foo>!
```

This approach is worked out along with the alternative for handling verbatims (see later on).

The \@checkoptions copes with repeated options via recursion by invoking \@checkoptions again. This process is stopped when no [is sensed.³⁸ It is complicated because the code has to look ahead for options and after that for the short-form separators or the end delimiter. And last but not least it is also complicated because the options must be processed with the right catcodes. When BLU dares to delve into the code he might perceive from \setupverbatim that the backslash has gotten category code 12 (other) and that therefore the commands supplied as optional arguments won’t work. (In the code \@SpecialsGetOther precedes \@checkoptions!) However, tugboat.sty’s comments say

```
‘... Since initial setup involves changing the special characters to characters of type other, some juggling must be done when optional commands are read. ...’
```

From the file we have ‘... \catcodes of \ { } are restored to their plain values ...’ via

```
\restorecat\
\restorecat\{
\restorecat\}
```

Too difficult for BLU, for sure.

And finally via \@executetoend the \curname @begin\CurrentTag\endcurname that is the \@begin<foo> macro is executed.³⁹

A poly-algorithm, and certainly not a UNIX filter to be combined easily with other filters. But uniform and consistent it is, for sure, if not for being so un-unusual.

2.2.7 Sectioning ‘commands’

Actually these are the markup commands for the headings. \head, \subhead, and \subsubhead all have

the same ending `\endhead`. (I have never used `\endhead` in practice, always the `\head*...*` form.⁴⁰)

‘Heads are set by first saving the text of the head in `\@argument`⁴¹ and then operating appropriately depending upon the `\headlevel`. Selection among the different heads is made by an `\ifcase`.’

The typesetting is done by `\@domainhead`, `\@dosubhead`, and `\@dosubsubhead`, for respectively `\head`, `\subhead`, and `\subsubhead`.

They all come basically down to typesetting of `\bf\the\@argument`.

Only the second (level) adds an end dot if `\if@headpunctuation` is true! (This default dot can be omitted via `\nopunctuation`.)

`\@next` is defined for each case and takes care of how the next line is separated: all (optional) spaces and `\par`-s are eaten, and only occasionally the main head does not allow for indentation. This `\@next` is general and also used after lists and verbatims. A neat convention!

I really can’t tell whether a head will pop up in print near the bottom of a page. For a (sub)subhead no penalties have been inserted so they might end at the bottom of a page. Not nice when fully-automated typesetting is strived after.

From the macro file I also stumbled upon

```
\sectitle...\endsectitle
```

with the underlying `\@sectitle` in `tugboat.cmn`. These are used by the editor to mark up ‘...the boxed title for the major column headings in *TUGboat*.’

2.2.8 Lists

Lists are preferably marked up by `\list... \endlist`.⁴² After the opening tag optional material might be specified, as usual. The options to choose from are what we call now the attributes: `\numbered`, `\romannumeraled`, `\Romannumeraled`, `\lettered`, `\Lettered`, next to `\ruled` and the specification of the tag `\tag{...}`.

Peculiar—but useful and I like these kinds of parameterizations—is the option

```
\def\itemseparator#1{\def\@itemseparator{#1}}
```

This will yield a comma as separator (default). However, this is only so when the list is *not* in the default `\displaystyle`. Confusing all those style variants, which by the way are not documented in the users’ guide.

TeXnically there is also the attribute `\itemized` (default) and `\unitemized`, where in the latter case the carriage-return separates the items. From the example in the users’

guide I get it that when the option `\numbered` is used one also has to be explicit about `\unitemized`. Hmm, looks error-prone to me. Reading the code reveals that my perception is wrong. With `\unitemized` the item label is updated and inserted via `\everypar`, where every end of line is also treated as an end of paragraph!

```
\everypar={\advance\itemnumber\@ne
            \tagform{\the\@itemtag}}
\makeCtrlMendgraf
```

Apparently too advanced or too much detail in order to be mentioned in the documentation.

The surrounding of the list by horizontal lines before and after is governed by the `\ruled` option.⁴³ For the moment there are no sublists supported.⁴⁴

The `\everylist` command is handy when one sticks to one kind of lists. For example, the default is a bulleted list and this can be changed into a numbered list via `\everylist{\numbered}`.

One can mimic L^AT_EX’s list. For example the `itemize` environment can be expressed in `\list...\endlist`, as follows.

```
\def\itemize{itemize}
\def\end#1{\ea
  \ifx\csname#1\endcsname\CurrentTag
  \endlist
  \else\message{No matching \CurrentTag
                end}
  \fi\egroup}
\def\begin#1{\bgroup\def\CurrentTag{#1}
  \ifx\CurrentTag\itemize\ea\list\fi
}%with LaTeX’s use
\begin{itemize}
\item ...
\end{itemize}
```

To mimic the other L^AT_EX list environments some more work needs to be done, especially the appropriate branching for, and the correct handling of, the specific situations.

The above has been included to illustrate the idea of a L^AT_EX user interface on top of `tugboat.sty`.

In order not to lose the wood for the trees I have neglected the `enumerate` and `description` environments.⁴⁵

The functionality provided by the optional parameters of the `\list` command is really handy for authors who change their copy often.

They will benefit most from the automatic numbering, be it by numbers or letters.

⁴⁰Note that because the form is parsed the syntax is flexible with respect to spaces around the `*`-s. This in contrast with the rigidity of the parameter separator TeXnique.

⁴¹Because of this verbatims will go wrong in the titles of the headings. L^AT_EX’s notorious fragile.

⁴²This holds especially when the `<Contents>` has to be marked up by it and not so much the `<Title>`.

⁴³This is something not special for lists. It can also be used with verbatims (or figures). The same with `\numbered`, to number lines in verbatim texts. However, I consider it better to provide this functionality separately and independently, as one of the ways we like to encapsulate document elements. Why not framed, to name but one alternative in use with tables?

⁴⁴Barbara Beeton commented that these are ranked high on her list of wishes.

⁴⁵Perhaps another time I will come back to it in full generality.

The coding of the options is a bit complicated because of the way alphabetic ‘numbering’ has to be coded in \TeX . The way to do this within \TeX is to bias the counter variable—`\itemnumber`—by "60, respectively "40, and advancing the counter as usual. With `\@itemtag={\char\itemnumber.}` the right letter will be obtained. A \TeX ing paradigm! Roman numerals go much along the same lines: `\@itemtag={\romannumeral\itemnumber}`.

Nested within `\list` is `\item` with optional parameter `\tag{...}`.

Alternative

Without optional parameters many alternative codings can be obtained via the use of Knuth’s `\item` enhanced with some code from `tugboat.sty`. The `\list` functionality—after abstracting from details—can be obtained via the following progressive coding. For the default, that is the bulleted list, the following.

```
\let\dekitem\item
\newtoks\itmtag \itmtag={\bullet$}
\def\item{\dekitem{\the\itmtag}}
\def\list{\prefoo\bgroup}
\def\endlist{\egroup\postfoo}
\def\prefoo{\def\postfoo{\%e.g.\hrule}}
%with use
\list
\item <item copy>
\endlist
```

The next step is to allow for ‘options.’ Below I have coded ‘`\numbered`.’ This can be added to the above. Similar things hold for the other options.

```
\newcount\itmnum
\def\numberedlist{\prefoo\bgroup
  \def\list{\itmnum0
    \itmtag={\global\advance\itmnum@ne
      \number\itmnum.}}
  \list}
%with use
\numberedlist
\item <item copy>
\endlist
```

Just the little extra to Knuth’s `\item`, to make our \TeX ing life a little easier. Not too clever, nor too general. Simple, straight and direct, respecting the way Knuth has shown us. The point I like to make is that more could have been built directly upon what Knuth had already provided.

Moreover, the checking of the correspondence of the `\begin<foo>` with the `\end<foo>` is no longer much needed since the advent of the (L^A) \TeX intelligent editors which prompt you with environments.

It is admitted, however, that `\everylist` does not combine easily with the above.

Intermezzo: prefix versus postfix.

Optional parameters are active between the opening and closing tag, with the option(s) supplied right *after* the opening tag. Loosely speaking this is reminiscent of ‘prefix operators.’ On the other hand my approach via the opening tag placed *after* the ‘option(s)’—as shown in the

above alternative—is reminiscent of ‘postfix operators.’ However, my mechanism yields after expansion a similar execution order. For the example above after expansion the execution processes goes along the lines

```
\prefoo%Whatever you wish to be done
\bgroup
\list%modified with numbered items
\item ...
\endlist%equals \egroup\postfoo
```

End intermezzo.

What confuses me are the more-column lists. What are we talking about? A table? Or do we just like to be economical with the space and cut a long list into pieces set next to each other? Is this descriptive markup? I never used this facility, and think I would not if only for tables. Is there a practical need for this?⁴⁶

2.2.9 Verbatims

Verbatims are provided functionally for

- in-line use, via `|...|`, or `\verbinline...\endverbatim`
- in display use, via `||...||`, or `\verbatim...\endverbatim`

with the following options

```
\inputfromfile{...}
\outputtofile{...}
\numbered
\continuenumbers
\ruled
\lastoption
\makeescape\...
\enablemetacode
```

The best verbatim functionalities (and codings) I have ever laid eyes on. Neat! Note that here a minimal form of markup can be used—by `|`, or `||` for begin and end tag for in-line and display, respectively—this while the `<Contents>` is enclosed by the begin tag and end tag. Next to that the general mechanism of supplying options can be used.

How are verbatims processed? The verbatims are processed via storing *all* the verbatim text until `\endverbatim`—or its alias—in `\@argument`. This is not easy to read from the code because first the opening tag—or its alias—and optional arguments are parsed via `TUGboat`’s mechanism. After that `\@executetoend` invokes `\@beginverbatim` with as replacement text `\obeyspaces` (`\obeylines` is on already.) Finally, in `\@longparse`—or `\@shortparse`—the verbatim text is stored in `\@argument` *with the right catcodes*. This code is advanced and a teaser for those who think they understand \TeX . For your convenience I have included below a piece of code—one of the parse variants—that takes care of the parameterization over the end separator.

```
1 \def\@longparse{\if@savingargument
2   \edef\@form{\def\nx\@longparse####1\the
3   \enddelim}%
4   \@long\@form{\@argument{##1}%
5   \cname end\CurrentTag\endcname}%
6 \else\def\@longparse{\}fi
7 \@longparse}
```

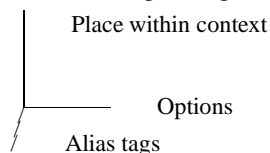
⁴⁶I get it that this feature is taken over from Berry’s explain?

Explanation. In lines 2&3 the concrete end delimiter is inserted via the expansion of the `\edef`. In lines 4&5 `\@@longparse` is defined with as replacement text the storing in `\@argument` of the verbatim text with the right catcodes (remember that `\setupverbatim` is on already) and the invocation of logically `\endverbatim` to typeset the (stored) verbatim text. Quite something isn't it?

Alternative: parameterization over end delimiter. This can be done differently. I used earlier⁴⁷ for parameterization over a separator the following two-level approach

```
\def\enddelim#1{\def\readtoenddelim
  ##1#1{\def\store{##1}}
%with use
\enddelim\enddelimitter
\readtoenddelim
%\store contains now all which follows
%after \readtoenddelim up to \enddelimitter
```

Intermezzo: functionalities. In general these kinds of functionalities as provided by `\verbatim` can be grouped into the following orthogonal sets.⁴⁸



By the way the above was hacked via

```
\setbox0=\vbox{\hbox{/}\kern0pt}
\def\xyz#1#2#3{\leavevmode \lower
  2.65\ht0\hbox{\copy0\rlap{\qqquad#1}}%
  \kern-.185\ht0\lower1.8\ht0\copy0%
  \kern-.185\ht0\lower0.95\ht0\copy0%
  \kern-.11\ht0%
  \vrule height9ex depth0pt\relax
  \raise7ex\rlap{\qqquad#3}%
  \vrule height.1pt depth.1pt width 8ex
  \qqquad\lower.6ex\hbox{#2}}
\quote
\xyz{Alias tags}{Options}{Place within context}
\endquote
```

Clarification

- place within context, denotes how the element is set within the context (in-line, display)
- options, denote the parameterization of the functionality proper (here: copy, file, ...)
- alias tags denote alternative tags (minimal markup, for example by `|`, or `||`).

The coding as provided is *not modular*. The functionalities are mixed up in the coding, also called monolithic or poly-algorithmic code. History has it that these kinds of codes are difficult to understand, to maintain, to extend, or to be ported (partially). If I had to code these sets of functionalities I would do that in an orthogonal way, each set independent from the others.

For example the ‘typesetting within context’ is not at all specific for verbatims, it applies equally well to other document parts like figures, math, tables, or you name it. Options—or let us call them parameters—are for example to have a display with rules before and after, or to have a frame around the document element as can be seen with tables. So these can better be provided separately in such a way that they can be used orthogonally with other document elements.

With respect to options I would—and have already in the past—adopt Knuth’s parameters. Here I played with the idea to provide `\thisverbatim`, in analogy with `\everyverbatim`, as is touched upon a little further.

I like to look at alias tags as extras which are virtually absent when not used. With verbatims there is a problem because of the fixed category codes. As shown in `tugboat.sty` one can parameterize over the end delimiter.

With respect to coding separate verbatim user macros for each functionality I consider that not economical because we like orthogonal combinations of the functionalities too. To provide codings for all these combinations would require too many macros.

End intermezzo.

Knuth’s verbatims. If we compare the above with the functionalities already provided in the `TEXbook`, Appendix D: 3 *Verbatim listing* then the additional mechanisms are

- the optional parameter mechanism provides choices to be made by the user (In The `TEXbook` Appendix D the file `verbatim` is numbered by default.)
- the writing to a file `verbatim`⁴⁹
- the definition of an escape character for semi-transparent verbatims.⁵⁰

However, in typesetting the `CWEB` manual Knuth&Levy introduced the following *verbatim mode*⁵¹

```
\def\verbatim{\begingroup\dospecials
  \parskip0pt \parindent0pt \let\!=!
  \catcode'\ =13 \catcode'\^M=13
  \tt\catcode'\ !=0 \verbatimdefs
  \verbatimgobble}
%with auxiliaries
{\catcode'\^M=13 \catcode'\ =13
  \gdef\verbatimdefs{\def^M{\ \par}%
  \let =\ }
  \gdef\verbatimgobble#1^M{}}
%with for example file verbatim via
\def\tt{\eighttt}\baselineskip9pt
\def\printmacs{\input cwebmac}
\verbatim
!printmacs
!endgroup%ends verbatim
```

⁴⁷ Locating a character in a string, or a card in a (bridge) hand, see my Syntactic Sugar paper.

⁴⁸ For the poor man’s slanted axis I first tried to box the / from the calligraphic font and was quite surprised to find a box of width 0!

⁴⁹ This has been coded in the `TEXbook` in the `\answer` macro. There the text will be written to `answer.tex`.

⁵⁰ This mechanism is handy with verbatims which contain pieces to be set in a font different from `\tt`, as worked out in the macro `\enablemetacode`. When this is used as option with `verbatim` the metacode can just be specified by `<metacode>` with result `(metacode)`. Handy!

⁵¹ With ! to be marked up via !!.

Agreed Knuth&Levy are by no means average T_EXies, but it gives food for thinking when a verbatim package is complex.⁵² In essence the structure is the same as with my `\thisverbatim`, differing modulo some syntactic sugar. Perhaps my approach is a nice compromise between simplicity and abstract rigorous markup, with as extra bonus no parsing overhead.

Another nice insight, due to Rainer Schöpf, is that verbatims and block comments are closely related. Block comments don't format, they just skip copy!

Intermezzo: block comment.

In practice I have used permanent and temporary block comments. The first is used for documentation (of code/macros), and are marked up in T_EX simply by starting each line with the %-character. The temporary comments are needed now and then to locate markup unbalances by commenting out document parts. A way how to do this is to let your editor insert %-s. Enclosing the document part(s) by `\iffalse` and `\fi` does not work when outer defs are enclosed (`\beginsection`, `\new<foo>`, ...). From a user point of view it is nice to provide `\comment` and `\endcomment`. Later on block comments will be treated as a special case of verbatims, so I'll refrain from targeted code.

End intermezzo.

A T_EXing paradigm is the parameterization over the end tag symbol the 'end verb delimiter.' First, we need '... to 'see' the tag when '\ ' is of type 'other'...' via

```
%In \setupverbatim
\enddelim=\ea{\endverbdelimiter}
%with the default
{\catcode'\|=0 \catcode'\|= \other
 |gdef|endverbdelimiter{\endverbatim}}
```

Second, we have the specifics for the `| |` (and similar for the `| |`). This has been accounted for via

```
\makevertverbchar%at end tugboat.sty
%which defines
\@verbchar={| |}
%and invokes
\setupverbchar
%the latter defines among others
\catcode\ea'\csname\ea\string
 \the\@verbchar\endcsname=\active
\edef\endverbdelimiter{\the\@verbchar}
\enddelim=\ea{\endverbdelimiter}
```

Verbatims seem to need 'processing on the fly' code. At the outer level this is so—no catcodes fixed—but at a lower level after the catcodes have been appropriately defined the verbatim <contents> is stored. This explains `\@savingargumenttrue`.

The big disadvantage of this is that long verbatims on small T_EX systems will yield the error message 'Capacity exceeded!'

⁵²If only I was aware of this when I submitted my first TUGboat article—Typesetting Bridge, when verbatim file inclusion was not yet available—my T_EXing life would have been simpler.

⁵³Or consult his article on the issue submitted to TUGboat.

⁵⁴Just a very minor issue, which can be settled via the escape character functionality.

⁵⁵This is characteristic for monolithic code.

In order to understand the various processes the reader is advised to begin with the T_EXbook p.380, where for example the details which come with inclusion of a file verbatim (with line numbering) has been discussed starting from

```
\def\listing#1{\par\begingroup
 \setupverbatim\input#1 \endgroup}
```

The cornerstone is of course `\setupverbatim`. For the coding of writing to a file verbatim the reader is referred to the `\answer` macro, which comes with `\exercise`, T_EXbook, p.422. Another educational inroad to the world of (file) verbatim intricacies is provided in Salomon's courseware.⁵³

In the TUGboat authors' guide the writing to a file has been illustrated for 'storing' address information—marked up at the beginning—and used at the end. In MHO—with all respect—this is a bit misplaced example, because this information could have been stored in token variables or so. More relevant examples deal with when moving information which needs catcode changes later on. Knuth did this with his `\answer` macro, his writing of index reminders, and I can think of the handling of endnotes as being relevant too.

So the conclusion is that Knuth's codings have been adapted to the general TUGboat coding schemes—especially obeying the orthogonal and systematic handling of optional parameters—next to the incorporation of some fine-tunings. I missed however the negative `\disablemetacode`—to switch it off temporarily—especially when we use the `\everyverbatim{...}`-s.⁵⁴

Alternative: the big deal. At the heart lies the wish to have verbatim functionalities available for use with AnyT_EX. As it is provided now it is too complex to be used within other contexts than `tugboat.sty`.⁵⁵

It is admitted that verbatim facilities are mostly needed by authors who write about (L^A)T_EX. The approach to classify facilities in orthogonal sets is fundamental though. So hang on. I'm not claiming that I have provided a production version, but I hope my approach will be recognized as being simpler. To illustrate my ideas I will treat below

- place within context, in-line versus display via `\preverbatim` and `\postverbatim`
- options, or parameterization via `\thisverbatim`
- alias tags.

In-line versus display. The whole idea is to concentrate on *one* verbatim functionality with markup structure

```
\verbatim
..
\endverbatim
```

that will handle them in general with no extras, just in-line, with `\obeylines on`, which is superfluous with really short verbatims, as a special case. The markup is always

in this form, with `\endverbatim` on a separate line. Then the only thing we need for displays is to enclose it by something like `\medbreak-s!` This can be parameterized eventually in `\preverbatim` and `\postverbatim`. Earlier I used this approach with respect to loops, see my Syntactic Sugar paper. This was what I had in mind, but in reality it worked out differently: first the display version and its variants, and later as a special case the (alias) in-line version. Not complete nor general. Sufficient and simple. Just what I need. Hang on.

Options via `\thisverbatim`. Basic is the idea of the toks variable `\thisverbatim` in analogy with Knuth's `\every<foo>`. To illustrate my approach I selected the following verbatim functionalities

- a user defined escape character for semi-transparent verbatims
- verbatims with lines numbered (TB, p.380, 381)
- input a file verbatim (TB, p.380).

First of all how would the user interface—the basis for any specs—look like? Is the following simple enough?

Examples of use.

```
%1. To handle metacode and font changes
\thisverbatim={\emc%enable metacode
\escapechar\!}

\verbatim
Some <meta code> and
blah, blah, ... !it
Now text in italics!tt
and back again in tt
\endverbatim
%
%2. To handle numbering and verbatim file
% inclusion
\everyverbatim={\numvrb}
\thisverbatim={\input \jobname.tex
\input vrb.tex}

\verbatim
Extras to be set (verbatim) after file
\endverbatim
%
%3. To handle block comments
\thisverbatim={\blockcomment}
\verbatim
Block comment
to be deleted/skipped.
\endverbatim
%
%4. To restart (line)numbers
\thisverbatim={\vrbline0 }%or supply \numvrb
\verbatim
Just some text with
line numbers restarted.

Text after two blank lines.
\endverbatim
```

For in-line verbatim I decided not to supply another version than via the alias tags because that is what is needed most of the times and simple. Hang on.

The coding alternative is based on TB p.382, modified with line by line processing via the fifo paradigm, van der Laan, 1993. The multi-level coding approach resembles the way how I have built the sorting macros, see my Sort-

ing in BLUe. The macros are available at the CTAN-s in directory `macros/generic/vrb`.

```
%Macro codes:
%User toks variables
\newtoks\thisverbatim
\newtoks\everyverbatim
%
%User Customing
\let\preverbatim\medskip
\let\postverbatim\medbreak
%
%User 'options'
%\numvrb
\def\numvrb{\vrbline0
\everypar{\advance\vrbline1
\llap{\sevenrm\the\vrbline\quad}}
%\blockcomment
\def\blockcomment{\def\processl##1{%
\def\preverbatim{}\def\postverbatim{}}
%\emc%my simplified enable metacode
{\makeactive\<
\gdef\emc{\makeactive\<
\def<##1>{\$ \langle##1\rangle$}}
%\escapechar\...
\def\escapechar#1{\catcode'#1=0 }
%
%User macro (\endverbatim implicit)
\def\verbatim{\preverbatim\begingroup\tt
\setupverbatim
\the\everyverbatim
\the\thisverbatim
\verbatimgobble}%\fifol for all but first
%
\def\verbinline{\bgroup
\def\preverbatim{\let\par\relax}
\let\postverbatim\egroup
\verbatim}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\def\makeactive#1{\catcode'#1=13 }
%
%Default
\def\processl#1{#1\par}
%
\newcount\vrbline
%
\def\setupverbatim{\makeactive\` \TB381
\def\par{\leavevmode\endgraf}\obeylines
\uncatcodespecials\obeyspaces}
%
{\makeactive\` \gdef'\{\relax\lq}}
%
\def\uncatcodespecials{\def\do##1{%
\catcode'##1=12 }\dospecials}
%
%For FIFO see TUGBoat 14, 1, EuroTeX '92
%proceedings, or MAPS 92.2
{\catcode'\|=0 \catcode'\|=12 \obeylines%
\gdef|ev{\endverbatim}%
\gdef|fifol#1
{|\def|tst|#1}%
|\if|tst|ev|lofif|fi%
|\processl|#1|fifol}%
\gdef|lofif#1|fifol{|\fi\endgroup%
|\thisverbatim{|\postverbatim}%
\gdef|verbatimgobble#1^M{|\fifol}%
|\obeyspaces|global|let =| }
```

Promising and simple. It provides most of what I need. Perhaps it should be worked out with all of the ornamenta as provided by `tugboat.sty`. For writing to a file verbatim—needed for writing index reminders, or answers to exercises—I would start with what Knuth already provided, ... and stay close to it.

What I like to get across is that the same functionalities can be obtained with less complicated coding, at least with more flexible code.

Alias tags. The opening tag is not a problem. And in general neither the closing tag. But with verbatims it is a bit difficult.⁵⁶ The specs could read

```
%5. In-line verbatim with alias tags
\thisverbatim{\emc}%enable metacode
Before |in-line <text>| after.
%with macros
{\makeactive\
 \gdef|{\bgroup\tt\setupverbatim
  \the\everyverbatim
  \the\thisverbatim
  \def|{\egroup%From manmac
    \thisverbatim={}}}
```

Hmmm, some work to be done. Given what is already available, it will have low priority. For the moment I stopped and will look over BLU's shoulder how things will go in practice.⁵⁷

2.3 Placement of figures

Placement of figures are accounted for via the markup

```
\figure
<vertical mode material>
\endfigure
```

with as result 'a single-column floating top-insertion.' The options implemented are

```
\top%default
\mid
\bot
\caption{...}
```

Peculiar is the provision of the caption via the token variable `\caption`, as option.⁵⁸

This stretches the option concept. Not only the representation is the subject of an option but also whether the components appear in print at all.⁵⁹

When two-column formatting is the default we need mechanisms to handle 'figures spanning columns.' With respect to the latter the following is borrowed from the users' guide.

'Figures spanning columns at the top and bottom of a page are currently supported only on the first page of an article, but we expect they will soon be allowed on any page (a general rewrite of the output routine is in progress). `\twocolfigure` (terminated by `\endfigure`) starts up such a figure and currently *must* occur before any material has been typeset on the first page (i.e. before `\article`).'

With respect to intermixing `\onecol`, `\twocol`, and `\threecol`, the following from the users' guide.

'... these can't currently be intermixed on a page.'

By the way the table about the functionalities of `(1)tugboat.sty` in the beginning of this paper, was set via

```
\figure[\mid]
$$\fll\htable\data$$
\endfigure
```

with `\data` and `\rowstblst` appropriately filled, and `\caption` of `\htable` adapted into `\capbtbl`, because of name clashes.

2.4 Argument processing on the fly

L^AT_EX's footnote first stores the footnote text while plain's footnote processes the 'argument' on the fly. In practical terms this means the we can use verbatims in plain's footnotes and not in L^AT_EX's.⁶⁰ I consider the coding of argument processing on the fly a T_EXing paradigm. The example of how to do this is provided in the T_EXbook by the coding of `\footnote`, p.363. That footnote coding buries the 'processing on the fly' T_EXing paradigm, because necessary details had to be accounted for, especially to allow one symbol or a group as 'argument.'

The need for this specific way of coding comes from the wish that

the catcodes of the 'parameter' at the time of processing must be different from the (permanently) assigned catcodes when parsed as argument.

Moreover, the catcode changes must be kept local. T_EXnically this comes down to the wish that `\foo{<text>}` must be processed as for example

```
\bgroup\x\y\x <text>\egroup
```

with `\x...z` any command, especially catcode changes. This simple case can be achieved by the following code.

```
\def\foo{\bgroup\x\y\z\let\dummy}
```

The paradigm is based on the reading away of the brace preceding `<text>`.

`\if@savingargument` indicates that a parameter will be processed on the fly in `tugboat.sty`. An example of `tugboat.sty` 'processing on the fly coding' is in `\figure... \endfigure`.

```
\figure <pic>\endfigure
%yields as essential replacement text
\topinsert %via \@checkoptions
<pic> %picture
\endinsert %via \endfigure
```

Another branch of the coding provides for processing the `<pic>` ture via

```
\setbox\T@stBox=\vbox\bgroup\hsize\pagewd
<pic>
\egroup
```

⁵⁶The eye-opener how to do this simple was supplied in `manmac`! Had been there from the beginning.

⁵⁷A white lie. I could not resist to finish it up with a verbatim mode suite of macros. See my BLUe's Verbatim.

⁵⁸Because my `btable` macro used also `\caption` I had to kludge by providing `\caption{...}` as optional argument within this context. That worked more or less. I decided to change the name into `\capbtbl`.

⁵⁹Within my `btable` macro I adopted when `\caption` is empty it won't be typeset. Reasonable right?

⁶⁰A modification of L^AT_EX's footnote in the T_EX spirit is provided in TTN2.4 by Jeremy Gibbons. It does not say that verbatims can't be used. No, but when special characters are used in verbatims they will be processed with probably unintended catcodes.

Explanation. As stated earlier `\@checkoptions` also executes to the end of the environment, meaning that it also invokes `\@beginfigure`. The latter codes two possibilities, the branches

- a fixed figure (and on the first page) starts with `\setbox\T@stBox=\vbox\bgroup\hsize\pagewd`, with the material in the input stream—the figure—to follow. On encountering `\endfigure`, `\egroup` is inserted, `\@caption` invoked and the page length adjusted. The latter via `\resetpagelgt`. The testbox is copied into one of the boxes `\firstf@ot` or `\firsth@ad`. The figure will then appear at the top (default), respectively bottom if the option `\bot` has been used.
- a floating figure is handled via the invocation of either `\midinsert`, `\botinsert`, or `\topinsert`, followed by the figure material from the input stream. It ends with `\endfigure`, which after `\@caption` inserts `\endinsert`.

Spanning figures have to be marked up by `\twocolfigure`, which has been defined as

```
\def\twocolfigure{\figure[fixed]}
```

Apart from the floating aspects of a picture it would be nice if `tugboat.sty` provides for a picture environment à la L^AT_EX. Why not take the relevant macros from `gkpmac`? If I place myself in the position to answer that question then I would say: ‘Some work has to be done to adapt these macros to the general coding philosophy.’ Hmm, not nice! Easy extension should not be hindered by the coding philosophy. I favor a practical and pragmatic extensible system.

2.5 Page make-up, or the OTR.

This is in general an advanced T_EX issue and can best be appreciated after chapter 23—and to a lesser degree chapter 15—of the T_EXbook have been understood. *TUGboat*’s output routine—OTR for short—is simple in the sense that it just uses `\box255`, also called `\@cclv`. The height of the text in `\@cclv` has been adapted in view of the floats to be added on the page in the OTR, via plain’s page breaking mechanism as detailed with in chapter 15 of the T_EXbook. To be concrete the amount of space the footnotes require has been accounted for. *TUGboat*’s OTR is complicated in the sense that it allows 1- and 2-column format, and that it allows for overlays to be handled either within T_EX or at the dvi-level.⁶¹ The latter means that pages are shipped out which contain the columns separately, to be pasted up later.

Add to this the quote from T_EXbook p.253

‘... Chapter 22 taught you how to be a T_EX Master, i.e., a person who can produce complicated tables using

⁶¹ Barbara Beeton communicated that the latter is ‘... an artifact of the alphatype typesetter, ... Rather than providing separate output routines in `tugboat.sty`, the proper thing is simply to eliminate this code that is no longer relevant. ...’ Agreed, wholeheartedly!

⁶² Even better as Barbara Beeton communicated ‘... there has been for a year or two an enhanced output routine in test form that permits switching (in the plain version) from two columns to one, and vice versa; this is at the moment rather tricky in use, not documented, and not at all ready to be made public; ...’

⁶³ Note added: the latter is an anachronism, and no longer relevant.

`\halign` and `\valign`; the following material will take you all the way to the rank of Grandmaster, i.e., a person who can design output routines. ...’

and the conclusion will be that the following is for hackers only. Furthermore, *TUGboat*’s OTR is under reconstruction.⁶²

Basically, a page can be shipped out via plain’s `\plain-output`, as follows

```
\shipout\vbox{\makeheadline
               \pagebody
               \makefootline}
\advancepageno%TB 252
\ifnum\outputpenalty>-20000 \else
  \dosupereject\fi
%with
\def\makeheadline{\vbox to 0pt
  {\vskip-22.5pt
  \line{\vbox to 8.5pt}{\the\headline}
  \vss}
  \nointerlineskip}
%and
\def\makefootline{\baselineskip24pt
  \line{\the\footline}}
```

If we look at the above for the headers and footers and those which actually appear in *TUGboat* then the conclusion is that the coding has been done with more complicated headers envisioned. For the moment plain’s simple approach will do, apart from the trim marks.

Plain’s `\pagebody` reads

```
\def\pagebody{\vbox to\size{%
  \boxmaxdepth\maxdepth\pagecontents}}
%with
\def\pagecontents{\ifvoid\topins\else
  \unvbox\topins\fi\dimen@=\dp255
  \unvbox255 \ifvoid\footins\else
  \vskip\skip\footins
  \footnoterule
  \unvbox\footins\fi
  \ifr@ggedbottom\kern-\dimen@\vfil\fi}
```

In the T_EXbook p. 257 it is detailed how to handle two columns. In `tugboat.sty` the columns are set in `\column1`, and `\column2`, and appropriately overprinted on the page.

So, I would suggest in order to make it simpler to provide two separate OTR’s: one for overlays to be handled within T_EX—then only the switching between 1- and 2-column has to be accounted for—and another to handle the overlays outside of T_EX, which ships out each column appropriately.⁶³ Better still is to provide a separate preprint style with a simplified OTR. For a proposal see Appendix C.

2.6 Back to TUGboat’s OTR

Let us forget about the anachronism of the alphatype typesetter and pass through the code to find out what is going on.

TUGboat's OTR is called `\output@`. This routine starts with

```
\@saveorship\midpage{\kern\coloffset
  \pagebody\hfil}
%with essentially
\def\midpage#1{\vbox{%space for runhead
  \vbox to\pageltgt{% runhead (or room for)
    \hbox to\pagewd{#1}%the column + offset
      % runfoot (or room for)
  }}
}}
```

which comes down to set 'columns'—via `\setbox`—and store these in `\column1`, `\column2` et cetera. `\pagebody`—in reality the invocation of `\pagecontents` in there—yields essentially the contents of `\box255`, as can be read from the above (plain) versions of `\pagebody` and `\pagecontents`.

After that `\output@` continues essentially with

```
...
\ifnum\xcol=\maxcols
  \shipout\hbox{\xcol\@ne
    \loop\rlap{\boxcs{column\number\xcol}
      \ifnum\xcol<\maxcols
        \global\advance\xcol\@ne
      \repeat
    \hbox to\pagewd{}}
  \else\global\advance\xcol\@ne
\fi
...
%with from tugboat.cmn
\def\boxcs#1{\box\csname#1\endcsname}
```

to ship out the page or to continue formatting, typesetting and storing the next column(s).

As can be expected `\newpage` has been adapted accordingly to cope with stored 'columns.'

```
\def\newpage{\vfill\eject
  \loop\ifnum\xcol>1
    {\leavevmode\endgraf
     \vfill\eject}% \xcol is
  \repeat} % advanced in OTR
```

The other problem of handling the last page is generally not an issue because articles may start at the end of the previous one. To assist the editor in this process some macros have been coded for 'page adjustment.' Too specific and too much detail.

Unfortunately, we have to wait a little for THE new OTR to come out, in order to switch happily from 2- into 1-column and vice versa anywhere on the page, for example to set page-wide tables, confidently.

2.7 Conclusion

A beautiful and handy set of tags have been provided for authors writing in \TeX about Any- \TeX . The coding of the general mechanism for tags is a marvel of \TeX ing, but a little hard to understand in its full generality.

Much attention has been paid to detail such as the automatic suppression of innocuous spaces and superfluous `\par`-s and blank lines.

I would like to see that `\<foo>*<head text>*` will be used to mark up the head, and

`\<foo><body text>\end<foo>` for larger chunks, with the heading set implicitly. The markup for keywords and abstracts can best be done by the latter. The checking of the corresponding `\begin<foo>` and `\end<foo>` can be omitted since the advent of (\LaTeX) \TeX intelligent editors. Together with the above suggested discipline of use—the environment is prompted by the intelligent editor—this can make the coding of the style (conceptually) much simpler.

Another royal road to simpler coding is to introduce `\this<foo>-s` next to the usual `\every<foo>-s`, instead of 'optional parameters.'

It would be nice to have the handling of verbatims separately available as a tool to cooperate with Any \TeX , especially \LaTeX .

The two distinct functionalities for the OTR must be (and will be) curtailed with respect to anachronisms. (Actually, the OTR will be replaced in due time.)

The uniform coding philosophy is an advantage with as the negative side that `tugboat.sty` is not easily extensible.

`tugboat.sty` was my style of choice to work with next to `manmac`. Because of the complex overhead and its incompleteness I will use simpler means to suit my purposes in future.

For example, for the markup of this article I used from `tugboat.sty` the functionalities

- abbreviations (from `.cmn` actually)
- title markup
- `\article... \maketitle \endarticle`
- major structure markup commands, like `\head*...*`
- verbatim functionalities in all its glory
- the default `\twocol` (the OTR).

I needed as extra

- `\keywords... \endkeywords`
- `\abstract... \endabstract`
- `\bitem` (bulleted items, and Knuth's `\item`, well I needed them because I like to be independent as much as possible from a specific style)
- `\quote... \endquote`
- (primitive) picture markup
- simple table markup (my `\btable`)
- bibliography handling (my BLUE's Bibliography)
- table of contents
- miscellaneous control sequences, such as `\cs` (to mark up a control sequence), `\em` (emphasize), `\partlogo` (from \TeX book), `\ftn` (with automatic numbering).

If I add my own markup commands for title handling, major structure markup commands, verbatim handling, `\multicolumn`—the hardest part, however, I don't need a sophisticated one—then I have the same functionalities, but simpler, faster, and more flexible. This emerged eventually in my proposal `tug.ppt`.

```
ltugboat.sty
```

3 LTUGBOAT.STY

For L^AT_EX submissions the T_EX Users Group relies on L^AT_EX's `article.sty`, and `art10.sty`. In the users' guide only one column is devoted to the use of `ltugboat.sty`. The version I inspected is 1.16e, Dec 92. Because this article is not about L^AT_EX adaptation this part is brief.

There is a problem in customing here. On the one hand we have the L^AT_EX parameters which can be adjusted and on the other hand we have `\hsize` and `\vsize` which are used by T_EX now and then automatically at a lower level.

From the L^AT_EX file—Output section—the parameters involved are given below.

- `\textheight`, height of text on page, excluding head and foot
- `\textwidth`, width of printing on page
- `\columnwidth`, in one-column the value `\textwidth`, and in two-column the value $(\textwidth - \text{columnsep})/2$.

Also `\@colht` and `\@colroom` must be initialized by the value of `\textheight`.

3.1 Customing

In order to let the appearance in print be similar to plain T_EX submissions some parameter settings have been altered.

The page size parameters involved are enumerated below.

```
\textheight=54pc
\columnwidth=18.75pc
\columnsep=1.5pc
\textwidth=39pc
%2\columnwidth+\columnsep=\textwidth
```

Headers and footers. Peculiar is

```
\def\PrelimDraftfooter{%
\dlap{\kern\textheight\kern3pc\rlap{\hbox
to\pagewd{\midrttitle\hfil\midrttitle}}}}
```

This 'footer' is invoked in

```
\@oddhead
```

along with the header material, while `\@oddfoot` has been left empty. Why? Weird!⁶⁴ Similar things hold for the 'even' counterparts.

Sectioning commands. The L^AT_EX commands are redefined with as replacement text the invocation of `\@startsection` supplied with the appropriate argument values, to yield a `tugboat.sty`-like result. For example

⁶⁴Barbara Beeton communicated '... It happens that there is a good reason for this. Placing both header and footer in the 'header' with a fixed distance between them ensures that this distance will never change. If the regular footer was used and the page overfull, the footer would move down by that overfull distance, sometimes even disappearing off the page. It is easier to diagnose an overprinted footer than an absent one, and the fact that a bottom-of-page element can be relied on always to be in exactly the same location provides an extra check on the general page setting, without having to look at the job log. ...' Hmmm, it seems a kludge to me, because of a deficient OTR. Nothing wrong with kludges, but mark them as such.

⁶⁵Note that the parameter is superfluous—like my late `\ftn`—it is just reinserted at the end.

⁶⁶However, a file verbatim style is available. See `\verbatiminput` in Schöpf, 1989.

```
\def\section{\@startsection{section}%
{1}{\z@}{-8pt}{4pt}%
{\normalsize\bf\raggedright}}
```

It is also stated

'Redefine style of section headings to look more like TUGboat. Start with redefinitions from `article.sty`. (Only `\section` correct so far).'

Funny looks

```
\def\abstract{\section{Abstract}}
\def\endabstract{}
```

It is a bit against the philosophy of

`\<foo>... \end<foo>`, which in `tugboat.sty` delimit the header, and for this case 'Abstract.' Now the word `abstract` has been wired in and has made this command language specific, and moreover `\<foo>... \end<foo>` should now enclose the *contents* of the abstract. (But because `\endabstract` is empty it can be placed anywhere!?) This cosmetics is confusing. I will just stay with L^AT_EX's

```
\section*{Abstract}
<Abstract text>
```

or redefine it with the original and general function, with use

```
\abstract
<Contents abstract>
\endabstract
```

As can be seen from this I'm not at all against this markup, but the code looked just so strange.

Footnotes. The only thing adapted is the appearance of the actual footnote via

```
\long\def\@makefnmark#1{\parindent#1
\noindent\hbox to2em\llap{\@makefnmark}
\null$\mskip5mu$#1}
```

As stated before the functionalities differ, especially L^AT_EX does not allow verbatims in footnotes (fragile!).⁶⁵

Lists. Adapted are the default values of L^AT_EX's lists as required by L^AT_EX in `\@listi`, `\@listii`, and `\@listiii`.

For example

```
\def\@listi{\leftmargin\leftmargini
\parsep=1pt plus 1pt minus 1pt
\itemsep\parsep
\listparindent=1em}
```

The latter macros are invoked in L^AT_EX's `\eval...` For customing L^AT_EX's lists see Goossens, Mittelbach and Samarin's paper on the issue.

Verbatims. Nothing is adapted here. So as long as L^AT_EX does not provide file verbatim this is not supported either in `ltugboat.sty`.⁶⁶

Figures. Here the title and number of the caption are customized via

```
\def\fnun@figure{{\bf Figure\thefigure}}
```

Bibliography. Essentially a section with (hardwired) title ‘References’ has been defined and the references that follow are formatted via (L^AT_EX’s) `\list`, as shown by the code below.

```
\def\thebibliography#1{%
  \section{References}
  \@mkboth{REFERENCES}{REFERENCES}}%
\list{[\arabic{enumi}]}{\settowidth
  \labelwidth{[#1]}\leftmargin\labelwidth
  \advance\leftmargin\labelsep
  \usecounter{enumi}}
\def\newblock{\hskip.11em plus.33em
  minus.07em}%\sloppy
\tolerance8000\hfuzz.5\p@\vfuzz.5\p@
\clubpenalty4000\widowpenalty4000
\sffcode\.\.=1000\relax
```

3.2 Conclusion

It is amazing that L^AT_EX’s `article.sty` could be adapted to mimic the appearance of `tugboat.sty` by so few commands. It is a pity that some necessary functionalities provided by `tugboat.sty` are not available in `ltugboat.sty`—like `file verbatim`—to guarantee that published macros are exactly those which have been used. With respect to non-English use the hardwired-ness of several English titles is unhandy.⁶⁷

L^AT_EX provides the command `\pagestyle` for page layout mods. This could have been used with let us say *tugboat* page style to be defined via `\ps@tugboat`. Why not? An example of how to do this is provided in `eurotex.sty`.

Barbara Beeton communicated that one of her priorities with respect to adaptation of L^AT_EX is to remove ‘...the inability to suppress the indent on the first paragraph after `\maketitle`.’

tugproc.sty

4 TUGPROC.STY

This customing of `tugboat.sty` is only some 350 lines. I inspected version 1.10, June 92.

The ‘Guidelines for the proceedings’ are nicely done. Useful are the inclusion of spelling conventions and information about font use. I inspected Guidelines version 1.06, May 93.

4.1 Customing

The adaptations concern mainly the title part and the back matter. For the title part the title is set left justified, followed by the complete author information, also left justified. Keywords are discouraged. The abstract is centered and spans two columns. No section numbering. After this the copy proper is set in two-column format as basis. The appendices start on a new page and are set in one-column.

⁶⁷ Agreed, *TUGboat* is in English.

⁶⁸ Note that therefore this command must follow the abstract in contrast with `tugboat.sty`’s `\article`.

The page size parameters involved have not been altered.

The headers and footers have a r(unning) header and footer. The title page of each article does not have a running head. The other pages contain on the even numbered pages the title and on the odd numbered pages the author name(s). The footers contain the issue information and the conference information (inbound), next to the page number (outbound).

`\article` takes care of the formatting of the title and abstract,⁶⁸ via essentially

```
\def\article{\setbox\startbox=
  \hbox to\colwd{
    \hbox to\pagewd{\vbox{%title}}
    \hbox to\pagewd{\hfil\vbox{
      %abstract head and material
      }\hfil}\vskip1pc}}\hss}
%
\twocolfigure\box\startbox\endfigure
}
```

Abstract. I like this `\abstract <abstract contents> \endabstract`. Simple and straight. Note, however, how the coding has been adapted to allow for `<abstract contents>` spanning the two columns. The command `\article` invokes `\theabstract`. The latter has been defined in `\endabstract` as can be seen from the definitions copied below.

```
\def\abstract{\@abstract[\longargument]}
\def\@abstract{\begingroup
  \def\CurrentTag{abstract}
  \@defaultoptions
  \@savingargumenttrue
  \@checkoptions}
%
\def\endabstract{\global\toks@=
  \ea{\the\@argument}
  \endgroup
  \edef\theabstract{\ignorespaces
    \the\toks@\unskip}}
```

Bibliography sets entries via

```
\head*Bibliography*
\entry{Laan C.G van der,
  'BLUES Bibliography'',
  MAPS 93.1, pages 205--210.}
%et cetera, with
\def\entry#1{\noindent\frenchspacing
  \hangindent\Hang#1}
\def\Hang{1em}
```

Appendices start on a new page in one-column format. The title of the appendix is set via

```
\def\appendix#1\endappendix{\newpage\onecol
  \centerline{\HEADfont#1}}
```

That this coding does not allow for the `*`-delimiters indicate that the editors did not want to adapt—or did not feel comfortable with—the general mechanisms available in `tugboat.sty`. I for one find the mnemonics `\endappendix` misleading. (With the regular end tags it does not hinder because then the `*-s` can be used.) It suggests the end of the appendix. From the ‘Guidelines’ the following with respect to the use of it.

```
\appendix Appendix\endappendix
\head*Spelling Conventions*
<Contents>
```

bf Coding: alternative. The general mechanism could have been applied, with automatically the flexibility with respect to the use of *-separators and the options which come with any head. (Similar to coding of \head.) However, I chose to keep it simple and to favor a name for the word appendix \nameappendix.

```
\def\nameappendix{Appendix}
\def\appendix{\newpage\onecol
\centerline{\HEADfont\nameappendix}
\DeleteOptionalSpacesandPars
{\noindent\ignorespaces}}
```

A user can customize \nameappendix, that is all. The user reads

```
\appendix
\head*Spelling Conventions*
<Contents>
```

This use of \appendix is different from the use of \keywords, \head and the like. It has a different function too.

4.2 The modification of the OTR?

Because the appearance of the page is different from regular issues of TUGboat—especially the first page—we expect that the OTR has been modified. In reality the structural modifications have been done via \article. The modification in \midpage is not essential. (The footnotes are overprinted, similar to the way it has been done in ltugboat.sty.)

4.3 Conclusion

The layout of the front matter and the back matter have been adapted. Also the page layout has been customized in agreement with the purpose.

Editorial boards can learn from this file how to customize tugboat.sty. However, the file also witnesses that it is difficult to remain consistent with the coding conventions already present.

Why not provide a tug.ppt style and shield authors from the differences?

ltugproc.sty

5 LTUGPROC.STY

This customing of ltugboat.sty is only some 200 lines. I inspected version 1.06a, Jan 93. Basically, the title runs over two-columns, the (complete) author information is left justified, and the abstract spans also two-columns and is part of the title. It suppresses section numbering (\setcounter{secnumdepth}{0}). and adheres to different parameter settings for white space around the headings.

The authors' guide is 'Guidelines for the proceedings,' the same as the one with tugproc.sty.

5.1 Customing

The page size is left invariant in correspondence with TUGboat.

The headers and footers have their own layout and contents via \@oddhead, \@oddfloor and there 'even' analogues. Customing is as follows, where the account for the suppression at the title page has been omitted

```
\def\@oddhead{\hfil\rm\rhTitle}
\def\@evenhead{\rm\rhAuthor\hfil}
\def\@oddfloor{\issue\hfil\thepage}
\def\@evenfloor{\thepage\hfil\issue}
\def\issue{MAPS 94.1}
```

The \dopagecommands is technical for the editors, I presume.

Section heads. Remarkable is the attention given to the markup of the abstract. A minipage centered after the title. \abstract provides the word 'Abstract' appropriately set, and starts a minipage and within that a list. \endabstract terminates the list and the minipage.

Appendices. To have appendices in one-column a \onecolumn command has to be inserted before each \section{Appendix: ...}.

Bibliography. In contrast with ltugboat.sty the L^AT_EX convention has been abstracted from.

```
\section{Bibliography}
\bibentry <name>, ``<title>''
<journal>, <pages>, <year>.
%et cetera
```

5.2 Switching into tugboat.sty

To get an idea of the differences in markup and the amount of work involved, I transformed the markup of my BLUe's Bibliography paper from ltugproc.sty into tugboat.sty.

The markup which had to be changed is supplied in the accompanying table.

ltugproc.sty copy	→	tugboat.sty copy
\documentstyle{ltugproc}	→	\input tugboat.sty
\begin{document}		
(ltug) title markup	→	(TB) title markup
\begin{abstract}	→	\abstract
\end{abstract}	→	\endabstract
\maketitle	→	\article %Before abstract
%No keywords	→	\keywords
		...endkeywords
\section*{...}	→	\head*...*
\subsection*{...}	→	\subhead*...*
\begin{itemize}	→	\u
\end{itemize}	→	\par
\item	→	\bitem %(essent.plain's)
\begin{quote}	→	\quote
\end{quote}	→	\endquote
\footnote	→	\ftn %with aut.numbering
\begin{verbatim}	→	\verbatim %(BLUe's Verb.)
\end{verbatim}	→	!endverbatim
\vrb	→	
\ldots	→	\dots
bibliography markup	→	... %(BLUe's Bibl.)
\end{document}	→	\makesignature
	→	\endarticle

Near all is reversible, except for the itemize part. It did take me an hour to convert this 6 page—2-column—article. Just change the style file? Forget it. The way out for authors is a common `tug.ppt` (preprint) style, to abstract from the regular and proceedings *TUGboat* issues.

5.3 Conclusion

An annoyance is that when using `ltugproc.sty` I have to remember that in the markup the `\maketitle` command must be supplied *after* the abstract, and not before as usual with L^AT_EX. This is because of the choice to have the abstract span two columns. (`\@maketitle` has been adapted.) I also have to remember that L^AT_EX's bibliography environment is not used. Too much detail to remember, so I will rely on my example template—empty article from last time—when I need `ltugproc.sty` again.

euro92.sty

6 EURO92.STY option

This style option has been designed by Rick Furuta in 1987, and used and adapted by Philippe Louarn for EuroT_EX '91, and by Petr Sojka for EuroT_EX '92.

Some may ask what does this have to do with TUG? Frankly, there is no direct link when looked at it from a formal point of view. However, I consider this also a descendant from the *TUGboat* styles, and more importantly it shows how to use L^AT_EX's `\ps...`, to customize for page styles. Next to that is my wish to have the styles I have to deal with discussed in one context.

The option is an adaptation to L^AT_EX's `article.sty` and some 290 lines long. `\@maketitle` has been recoded to account for the various white spaces, the `\hrule`, and the fonts used. The headings are customized with respect to the contents of the arguments and also to account for the settings of the surrounding white space. There is no version indication included nor a history of changes.

The 'Guidelines for authors' has been done nicely. L^AT_EX authors can just start from the template—the guide itself—and obey the supplied information. T_EX authors are requested to obey the included specifications. Good! Perhaps it is too much to ask from a T_EX author to obey the specifications. The temptation to ride one's hobby in details is always there.

6.1 What does the EuroT_EX proceedings look like?

Basically a table of contents with the sequence of articles next. It is essentially one-column biased. In each article the title is underlined by a `\hrule`. The author information is set left justified, followed by the (centered) abstract and (centered) keywords (in one or two languages). Then the sections follow. Appendices are at the end. The headers on the pages have the author name (in italics and underlined with a page-wide `\hrule`) on the left-hand pages,

and the title of the article (in italics and underlined with a page-wide `\hrule`) on the right-hand pages. As exception the first page of each article has the information about the conference (in roman and underlined again with a page-wide `\hrule`). The footers contain the page numbers outbound.

The typographic quality of the proceedings I have seen is moderate, with all respect. Trivial errors in there. English proof readers could have contributed too.

6.2 Customing

The page size parameters are as follows

```
\textheight=546pt
\textwidth=12.7cm
%and some others involving \parskip and
%various separators
```

Headers and footers are controlled by

- `\def\ps@titre`, for title page style (with the conference information appropriately formatted)
- `\def\ps@gut`, with `\@oddhead`, `\@oddfont`, and their even analogues, appropriately defined.

The author has to provide the information for the running heads in

```
\titlehead{<short title for paper>}
\paperhead{<author name(s)>}
```

At the end of the style file `\pagestyle}{gut}` and `\thispagestyle{titre}` take care that the appropriate information will be used.

The L^AT_EX macro `\@outputpage` has been modified to include the underlining of the headers via the insertion of

```
\vskip10pt
\hbox to\textwidth{\hrulefill}
```

For the markup of other material just use L^AT_EX.

Intermezzo: abstraction from def versus toks variable.

At the user level one can't tell from for example

```
\titlehead{<Short title for paper>}
```

whether this is implemented as a toks variable or a (second level) definition.

```
\newtoks\titlehead
%or, as is the actual case
\def\titlehead#1{\gdef\@titlehead{#1}}
```

The advantage of introducing the second level defs is that the user doesn't have to think of the `\def` token. For a novice there is confusion.

End intermezzo.

6.3 Conclusion

This style shows how L^AT_EX's OTR—or more specifically `\@outputpage`—can be customized. The Guidelines for authors is in itself a nice template for an author's article. Adaptation via `\ps@...` is well-done, and that is the way how it must be done.

The bad news is—I'm sorry to say so—that the resulting typographical quality has been moderate.

7 TTNXNX.STY

This short style for TTN is based on L^AT_EX's report, and essentially 1-column, apart from the index. It contains its own abbreviations.⁶⁹ It checks whether the NFSS is in use and if so it takes appropriate action. Each major part has its title centered enclosed by `\hrules`. Sections have the titles left and author information flushed right. I inspected `ttn2n3.sty` of 1993.

For submissions of NTG's meeting reports I could just submit the ASCII copy, and the editor—Christina Thiele—inserted the few markup commands en-passant, while polishing my use of English.

7.1 Customing

The page size parameters involved are

```
\textwidth=29pc
\textheight=43pc
\voffset=-2pc
\overfullrule=0pt
\hfuzz=5pt
```

The headers and footers have a r(unning)-title and from the style I can't find how they were included.⁷⁰

Title parts are redefined as follows

```
\newcommand{\Section}[1]{\section*\centering
  \hrule\hrule\vskip.5pc{\Sectionfont #1}
  \vskip.5pc\hrule\hrule\vskip1pc}
\squasheadsubsection{foo}{bar}
%gives similar results to
%\subsection*{foo}\vspace*{-1.5pc}
%\begin{flushright} bar\end{flushright}
%\noindent
```

Too much detail and too specific to discuss the code here.

7.2 Conclusion

The newsletter is and looks great. Again, to my knowledge, no formal specifications are there for the layout of TTN. They just grew.⁷¹

— * —

8 Looking back

The style files are a rich source and the documentation is well-done. The coding of `tugboat.sty` is superb, although unnecessary complicated now and then. Nevertheless, I learned a lot from them. But this monolithic

way of coding is not my style. I prefer orthogonality and modularity.

Given that many people have been involved at geographically widely spread places—and worked under the pressure of continuous lack of time—I'm happy to conclude that the amount of inconsistencies is low, very low, and that the overall quality is good, although a bit difficult to read. But that is in the nature of T_EX being so unusual, I presume.

It is useful to take over the abundant good ideas which popped up at many places in the styles. As a non-T_EX specific example one can think of the 'spelling conventions' useful for non-native speakers of English. These should be made separately available to grow and to be generally used.

In working on this article I realized more and more the specific function of keywords and abstract, next to other bibliographic information. Because of this it is worthwhile to typeset these differently—I chose a smaller type—from the copy proper.

For optional parameter handling as done in `tugboat.sty` a much simpler alternative is to make use of `\this<foo>`, analogous to `\every<foo>`.⁷²

If we compare the `amsppt.sty` with `tugboat.sty` then I conclude that the handling of optional parameters goes much along the same lines. Furthermore, `amsppt.sty` is more developed with respect to the formatting of math, and `tugboat.sty` more with respect to handling verbatims.

Styles with a rigid and advanced coding approach tend to become cost-intensive to maintain, especially with adaptation

- to changing circumstances—the example in the past has been the (user) migration from (flexible) plain into (rigid) L^AT_EX, the latter has been under revision for the last 5 years—or
- to extending it with useful functionalities developed elsewhere under a different coding philosophy.

This phenomenon is general.

In the past the software engineers have experienced similar difficulties with big and rich computer languages—e.g. ALGOL 68, ADA, . . . , and not to forget huge operating system (as opposed to the RISC approach)—which have given birth to the phenomenon called 'little languages.'

⁶⁹This demonstrates the need for a separate file for abbreviations to be used within any context.

⁷⁰Christina communicated '. . . They're inserted into the source `.tex` file, not in the `.sty` file.

⁷¹Christina Thiele communicated 'The same, more or less, can be said about the *TUGboat* and proceedings macros: there has *never* been enough time to make sure that the macros are all done nicely, to agree with philosophical and overall consistency issues. They just grew, because over time, different situations arise, and it's more important to solve the immediate problem than to go back through the entire set of macros, to make sure that everything is still consistent and looks nice.' Fair enough. So be it. Like monolithic coding this *continuously* working 'behind the time' is not my style either. If I would experience *continuously* working behind the time, bells would ring that something is wrong there, it should be rearranged and in general I would face the choice '*dare to do less*,' as communicated at the Aston BoD meeting.

⁷²This applies also to `amsppt`'s style.

There are always quibbles from outsiders like me. I hope nevertheless that my remarks are well-taken and/or will assist people in understanding what is going on. The very least is that my reflections show how the styles are perceived by BLU.

The TeXing paradigms I stumbled upon are summarized below.

- data abstraction—or hiding the `\def` token⁷³
- alphanumeric numbering
- optional parameter handling
- outer def-s as argument of a test by hiding the test in another layer
- allow for either *-separators or an end delimiter
- ‘parameter separator’ with optional spaces before and after, via parsing
- minimal user markup for verbatim, next to complete `\verbatim...\endverbatim`
- flexible escape mechanism in verbatim
- end verbatims via `|<escape character>egroup|`
- parameterization over parameter delimiters
- processing on the fly of ‘arguments.’

I introduced myself the concept of `\this<foo>`.

9 Looking forward

Software engineering has it that for complex projects the concept of proto-typing is used. A realization of this is that when software is designed and coded it will be thrown away, and with the knowledge gained during the project new specs and code will be developed. To quote from Heckel

‘Prototype, revise, and rewrite.’

Pondering about the above in relation with the TUG style files, the following came to mind.

I would first like to see the specs—purpose, specifications, examples of use—and a set of requirements to which the code should obey. Then I would ask myself the (rhetorical) question whether the *TUGboat* styles should phoenix.⁷⁴ I mean redesigned with simplicity, flexibility, generality, extensibility and robustness as yard sticks, in the form of a literate program. This entails that the development of the code goes hand in hand with the emerge of the documentation. The extra bonus of this approach is the immediate availability of an index and the change file concept as tool for customing.

Perhaps the above idea is already outdated in view of the hypertext developments. What about ‘Style files as an hypertext,’ to paraphrase Mary Dyson?

Acknowledgments

Barbara Beeton helped me with recent copies of the style files and with some comments as well as with her lecture at the SGML-TeX meeting, 1990, at Groningen. Next to that

she proofed the article and pointed out several misconceptions, and made suggestions for improvement. Christina Thiele polished again my English.

Włodek Bzyl contributed to a clearer description of the ‘generic coding pitfall,’ next to clarifications of the used terminology, and pointing to the `gkpmac` format used for typesetting ‘Concrete Mathematics.’ Moreover, when I talked to him about my ideas to continue my BLUES trilogy with *TUGboat* BLUES, he pushed me forward a bit, and also challenged me to embark upon literate programming.

Ron Whitney read it nearly all through and put things within the right historical context, next to pointing to some vagueness now and then, especially in the `tugboat.sty` part.

Thank you! Any inaccuracies left are oversights on my part. I welcome constructive comments. History has it that ‘flames’ are the valuable ones, although wrongly packed. Comments which point to my abuses of English—especially the blind spots—are welcomed in particular.

Bibliography

- [1] Beeton B.N (1990): *TUGboat* production: TeX, LaTeX, and paste-up. SGML-TeX meeting, Groningen. (Unpublished)
- [2] Bzyl W, T Przechlewski (1993): An application of literate programming: creating a format for the Bulletin of the Polish TUG. TUG ’93. *TUGboat* 14, no. (3), 296–299. (Very promising this using of the experience embodied in `tugboat.sty` via transformation into WEB and modify this with GUST’s mods via change files. As a bonus there is an index. Handy to locate commands, but it won’t help to locate the various `\@begin<foo>-s` commands because these are invoked via `csnames`.)
- [3] Dyson M.C (1992): The curriculum as a hypertext. EP-ODD, 5, 2, 63–72.
- [4] Gibbons J (1993): Footnotes with verbatim material. TTN, 2, 4, p.9. (LaTeX’s footnote implementation has been modified to allow also for verbatims similar to plain’s coding. Basically it comes down to handling the ‘argument on the fly.’ Really nice.)
- [5] Goossens M, F Mittelbach, A Samarin (1993): The LaTeX-companion. Addison-Wesley.
- [6] Goossens M, F Mittelbach, A Samarin (1993): customing LaTeX’s lists. MAPS 93.2, 177–183.
- [7] Hamilton H (1989): Mastering TeX with templates. TUG ’89. *TUGboat* 10, no. (4), 541–548.
- [8] Heckel P (1982): The elements of friendly software design. Warner Books. ISBN 0-446-38040-7.
- [9] Knuth D.E (1984): Computers and Typesetting. The TeXbook. Addison-Wesley. ISBN 0-201-13447-0 (hard cover) ISBN 0-201-13448-9 (soft cover). (For the correct printing look in the index for `\language` or `\emergystretch`.)

⁷³From `\foo<copy>` a user can’t tell whether a toks variable has been used or a (second level) `\def`.

⁷⁴Ralph Youngen from AMS communicated to me at Aston that AMS is thinking about abandoning the `\nofrills` methodology altogether.

- [10] Knuth D.E, S Levy (1987): The CWEB System of Structured Documentation. (FTP: labrea.stanford.edu, in directory /pub/cweb)
- [11] Laan C.G van der (1991): Math into BLUes. Part I: Mourning. TUG '91. *TUGboat* 12, no. (3), 485–501. Part II: Sing your song. EuroT_EX '91. GUTenberg Cahiers 10&11, 147–170. (Pre-release MAPS 91.1.)
- [12] Laan C.G van der (1992): Tower of Hanoi, revisited. *TUGboat* 13, no. (1), 91–94. Also MAPS 92.1.
- [13] Laan C.G van der (1992): Table Diversions, MAPS 92.2, 115–129. (An earlier version at EuroT_EX '92.)
- [14] Laan C.G van der (1992): FIFO and LIFO sing the BLUes. *TUGboat* 14, no. (1), 54–59. (An earlier version at EuroT_EX '92, 225–234, and MAPS 92.2.)
- [15] Laan C.G van der (1992): Syntactic Sugar. TUG '93. *TUGboat* 14, no. (3), 310–318. (Earlier versions in MAPS 92.2, abridged in GUST bulletin 1.)
- [16] Laan C.G van der (1993): Manmac BLUes—or how to typeset a book via T_EX. MAPS 93.1, 171–191.
- [17] Laan C.G van der (1993): AMS BLUes—professionals at work. MAPS 93.1, 192–212.
- [18] Laan C.G van der (1993): Sorting in BLUE. MAPS 93.1, 149–170. (Abridged TUG '93. *TUGboat* 14, no. (3), 319–328.)
- [19] Laan C.G van der (1993): BLUE's Bibliography—a generic approach. MAPS 93.2, 205–210.
- [20] Laan C.G van der (1994): BLUE's Transparencies—From report to transparency. MAPS 94.1, 111–114.
- [21] Laan C.G van der (1994): BLUE's Verbatim—The selection MAPS 94.1, 116–118.
- [22] Lamport L (1985): *L^AT_EX User's Guide & Reference Manual*. Addison-Wesley. ISBN-0-201-15790-X. (With respect to BibT_EX the following characteristics from UKTUG meeting of 1990. Advantages: clear layout of database, unique identifier for elements, compatibility with Scribe databases, explicit statement of citation type, extensible style language, uses L^AT_EX's cross-referencing, easy to edit output, can be mixed with non-automatic generated bibliographies, cross-referencing and abbreviations. Disadvantages: multiple passes (L^AT_EX, BibT_EX, L^AT_EX, L^AT_EX); have to remember unique references, style language is opaque, database is very wordy and boring to enter.)
- [23] Louarn P (1991): Instructions to EuroT_EX and GUTenberg '91 authors.
- [24] Mittelbach F (1989): An environment for multi-column output. *TUGboat* 10, no. (3), 407–415.
- [25] Mittelbach F (1989): The doc-option. *TUGboat* 10, no. (2), 245–273.
- [26] Mittelbach F, R Schöpf (1989): The new font selection scheme. Reprint: *TUGboat* 11, no. (2), 297–305. (See also NFSS 2, in Goossens, Mittelbach Samarín.)
- [27] Mittelbach F, C.A Rowley (1993): The L^AT_EX3 project. MAPS 93.1, 95–99. (Also in various early nineties proceedings.)
- [28] Rahtz S.P.Q, M Burbank (1993): Guidelines for Proceedings of the 1993 Annual Meeting of the T_EX Users Group. TUG Office.
- [29] Salomon D (1992): NTG's Advanced T_EX course: Insights and Hindsight. MAPS Special, ≈ 500p.
- [30] Schöpf R (1989): A new implementation of the L^AT_EX verbatim[*] environments. *TUGboat* 11, no. (2), 284–296.
- [31] Spivak M.D (1989): *L^AM_S-T_EX—The Synthesis*. T_EXplorators Corporation.
- [32] TUG (publications): *TUGboat*, TTN, T_EXniques. (*TUGboat*: Scholarly quarterly of TUG, with the proceedings of the annual meeting as special issue, and the resource directory as supplementary issue. From 1980 onward. The style files involved are `tugboat.cmn`, `tugboat.sty`, and `ltugboat.sty` available from the (CTAN) archives. T_EXniques: Special themed editions, with no 1 to 14 of 1993) TTN: A portable quarterly newsletter of TUG. From 1990 onward.)
- [33] Whitney R.F, B.N Beeton (1989): *TUGboat authors' guide*. *TUGboat* 10, no. (3), 378–385. (Updated versions via the file server.)

— * —

Appendix A: Templates

In the templates I incorporated a simple and characteristic way of markup for the title part, sections, footnotes, lists, verbatims, figures, bibliography and appendices. From this a reader can easily 'jump off' from the template in the spirit of the work of Hope Hamilton.

The anthology shows that the markup is a little varied.

It is hoped in making the diverse markups explicit developers will realize that nearly the same functionality can be reached with less, much less variation. Syntactic Sugar? Yes!

A.1 Template: `tugboat.tem`

```
%tugboat.tem for .sty version 1.14
\input tugboat.sty %\input tugboat.cus
%
\def\runtitle{TUGboat BLUes}
\title*\runtitle---{\rm how \TeX ies do it}*
\author*Kees van der Laan*
\address*Hunzeweg 57, 9893PB, \dots*
\netaddress*cgl@riscl.rug.nl*
%
\article
%\head*{Keywords}* I use the mod
\keywords tugboat.sty, \dots
\endkeywords
%\head*{Abstract}* I use the mod
\abstract Abstracts and keywords
are special. I use my variants.
\endabstract
%
\head*Introduction*
Blah, blah and more blah \dots
\head*Sectioning commands*
Nothing special, just |\head*...*|,
|\subhead*...*| and |\subsubhead*...*|,
to be followed by the contents of the
sections.
```

```

%
\head*Footnotes*
Text with verbatim in
footnote.\footnote*{|Verbatim| text.}
%
\head*Lists*
\list
\item Note |\item| has been redefined
  within |\list...\endlist|.
\endlist
%
\head*Verbatims*
I love |\foo{<argument>}|, possible via
|\enablemetacode| option.
%
%Tables? Generic, via my \btable?
%Math? See the TeXbook
%Symbolic cross-referencing?
% (See Spivak or my Math into BLUes)
%
\head*Figures*
Floating objects via
||\fig<vertical material>\endfig||
%
\head*Bibliography*
Not supported other than plain's
facilities. I use
\item{[1.]} Laan C.G van der (1993):
  BLUes Bibliography. MAPS 93.2, 205--210.
%
\head*Appendix A: Templates*
Appendix, nothing special.
\makesignature
\endarticle          %cgl@riscl.rug.nl

```

A.2 Template: ltugboat.sty

```

%ltugboat.sty for .sty version 1.16e
\documentstyle{ltugboat}
\begin{document}      %\input{ltugboat.cus}
\def\runtit{Sorting in BLUe}
\title{\runtit}
\author{Kees van der Laan}
\address{Hunzweg 57, 9893PB,
         Garnwerd, The Netherlands,
         05941--1525; cgl@riscl.rug.nl.}
\date{}
\maketitle
%
\subsection*{Keywords:}Sorting, \ldots
%
\section*{Abstract}
'Abstract' according to language.
%
\section*{Introduction}
%Copy proper a la LaTeX
%
Text\footnote{No verbatims!}
%
\begin{itemize}
\item ...
\end{itemize}
%
%Tables? Generic via my \btable?
%Math? See the TeXbook
%
\begin{figure}
\hbox{...}
\caption{...}
\end{figure}
%Back matter
\begin{thebibliography}{abc}
\frenchspacing
\bibitem{cgl}Laan C.G van der (1993):
Sorting in BLUe. \tubissue{14}{3),

```

```

319--328.
(Unabridged MAPS 93.1, 149--170.)
\end{thebibliography}
%
\appendix
\section{Heap sort}
\makesignature
\end{document}          %cgl@riscl.rug.nl

```

A.3 Template: tugproc.sty

```

%tugproc.sty for .sty version 1.10
\input tugproc.sty    %\input tugproc.cus
%
\def\runtit{Manmac BLUes}
\title*\runtit---{\rm
  or how to typeset a book via \TeX}*
\author*Kees van der Laan*
\address*Hunzweg 57, 9893PB, \dots*
\netaddress*cgl@riscl.rug.nl*
%
\abstract The manmac macros are \dots
\endabstract
%
\article%Sets title and abstract
% \head*Keywords*CAT, ...%Discouraged
%
\head*Introduction* Nothing special here.
%
\head*Sectioning commands*
Just |\head*...|, |\subhead*...| and
|\subsubhead*...|, and contents.
%
\head*Footnotes*
Text with verbatim in
footnote.\footnote*{|Verbatim| text.}
%
\head*Lists*
\list
\item Note |\item| has been redefined
  within |\list...\endlist|.
\endlist
%
\head*Verbatims*
I love |\foo{<argument>}|, possible via
|\enablemetacode| option.
%
%Tables? Generic via my \btable?
%Math? See the TeXbook
%Symbolic cross-referencing?
% (See Spivak or my Math into BLUes.)
%
\head*Figures*
Floating objects via
||\fig<vertical material>\endfig||
%
\head*Bibliography*
\entry{Laan C.G van der (1992):
Table Diversions.
Proceedings Euro\TeX '92, 191--211.
(Adapted MAPS 92.2, 115--129.)}
%
\appendix Appendices\endappendix
\head*A: The file manmac.tex*
Page wide text.
%No signature!
\endarticle          %cgl@riscl.rug.nl

```

A.4 Template: ltugproc.sty

```

%ltugproc.sty for .sty version 1.06a
\documentstyle{ltugproc}
\begin{document}      %\input{ltugproc.cus}
%
\def\runtit{Sorting in BLUe}

```

```

\title{\runtit}
\author{Kees van der Laan}
\address{Hunzeweg 57, 9893PB, \ldots}
\netaddress[\network{%
  Internet}]{cgl@riscl.rug.nl.}
\begin{abstract}
'Abstract' according to language.
\end{abstract}
\maketitle
%
\section*{Introduction}
%Copy proper a la LaTeX
%
Text\footnote{No verbatims!}
%
\begin{itemize}
\item ...
\end{itemize}
%
%Tables? Generic via \btable
%Math? See the TeXbook
%
\begin{figure}
\hbox{...}
\caption{...}
\end{figure}
%Back matter
\begin{thebibliography}{abcdef}
\frrenchspacing
\bibitem{cgl}Laan C.G van der (1993):
Sorting in BLUe. \tubissue{14}(3),
319--328.
(Unabridged MAPS 93.1, 149--170.)
\end{thebibliography}
%
\appendix
\section{Heap sort}
\end{document} %cgl@riscl.rug.nl

```

A.5 Template: eurotex.tem

Below the author name must be supplied twice because of the definition

```
\def\authorhead#1{\gdef\@authorhead{#1}}
```

If a toks variable was used instead then we had a @-command less and we could use the toks variable in \title too, as follows.

```

\authorhead={<author name>}%= optional
\title{\the\authorhead}

```

The reason I can think of is the *overuse* of @-commands. In this case the @-command is completely superfluous. In general it is not clear to me either when to use toks variables and when to use defs. One case where it is extremely clear is when too many toks variables are needed (≥ 256).

```

%eurotex.tem for euro92.sty
\documentstyle[euro92]{article}
\begin{document} %\input{eurotex.cus}
%
\title{Table Diversions\thanks{A little
  different from proceedings Euro\TeX\ '92,
  especially in the coding of FIFO.}}
\titlehead{Table Diversions}
\author{Kees van der Laan}
\authorhead{Kees van der Laan}
\affiliation{Hunzeweg 57, 9893PB,
  Garnwerd, The Netherlands,
  05941--1525; cgl@riscl.rug.nl.}
\maketitle
%
\begin{abstract}Characteristics \ldots

```

```

\end{abstract}
\begin{keywords}{Bordered) Tables, ...}
\end{keywords}
%
\section*{Introduction}
%
%For other material a la LaTeX.
%
\begin{thebibliography}{abc}
\bibitem{cgl} {Laan C.G van der
  (1992): Table diversions.
  Proceedings Euro\TeX{} '92, 191--211.
  (Adapted MAPS 92.2, 115--129.)}
\end{thebibliography}
\end{document} %cgl@riscl.rug.nl

```

A.6 Template: ttn2n3.tem

```

\documentstyle{ttn2n3}
\begin{document}
\pagestyle{empty}
%
\begin{center}
{\Sectionfont\TeX{}} and TUG NEWS
\end{center}
\noindent \TTN\ is a newsletter for
\TeX{} and \LaTeX\ users alike\ldots
\pagestyle{myheadings}
\markboth{Vol.\~0, No.\~0, May 1991\qqquad
  {\TTN}}{\TTN}\qqquad Vol.\~0, No.\~0,
  May 1991}
%
\setcounter{page}{1}
\Section{Editorial}
%
\squashedsubsection{Welcome to \TTN!}{%
  Christina Thiele \\\
  Editor, Prototype \TTN \\\ May 1991}

\squashedsubsection{The Truth about \TeX}{%
  Christina Thiele\Carleton University}
%
\noindent Have you ever run across
articles or descriptions of \TeX{\ldots}

%specific items: Upcoming Events,
% TTN index (2-column)
\end{document}

```

Appendix B: Customing

The main purpose of this section is to show the <style>.cus files which account for your own page size, running headers and footers.

B.1 Customing: tugboat.cus

```

%tugboat.cus Dec 93
\hoffset-.75cm\voffset-.5cm
\normalcollgt=25cm\collgt=\normalcollgt
\pagewd=18cm \twocolcolwd=8.75cm
\intercolwd=.5cm
\resetpagelgt \twocol \pageno1
\enablemetacode
\everyverbatim{\enablemetacode}
\nopunctuation\overfullrule0pt
%
\def\rtitle{\hbox to \pagewd{\small
  \issue\hfill{\it\runtit}}}
\def\rfoot{\hbox to \pagewd{\tiny
  \rlap{Draft \today}\hfill
  -\thepage-\hfill
  \llap{\copyright cgl}}}
\def\issue{MAPS 94.1}

```

```
%with
\let\small\sevenrm \let\tiny\fiverm
\let\thepage\folio
\let\ea\expandafter \let\nx\noexpand
\let\ag\aftergroup
%and macros
\def\keywords{\subhead*\sevenbf
Keywords: *\bgroup\small\baselineskip9pt}
\def\endkeywords{\smallskip\egroup}
%
\def\abstract{\centerline{\sevenbf
Abstract}\bgroup\quote\small
\baselineskip9pt}
\def\endabstract{\endquote\egroup}
%
\def\quote{\endgraf\bgroup\narrower
\smallskip\noindent}
\def\endquote{\smallskip\egroup\endgraf
\noindent}
%
\newcount\fcnt
\def\ftn{\advance\fcnt1
\footnote{$\{ \}^{\the\fcnt}$}}
\def\verb{}
\def\cs#1{\tt\char92#1}
\endinput %cgl@risc1.rug.nl
```

B.2 Customing: ltugboat.cus

With respect to the page parameters I could have used `a4.sty`. Because I wanted full flexibility—not restricted to L^AT_EX's way—and that also the running headers and footers need adaptation I just inserted what is needed. Perhaps I should have used `\ps@cgl`? I refrained from that, because of too early generalities.

```
%ltugboat.cus Dec 93
\hsize8.5cm \vsize25cm
\overfullrule0pt
\textheight\vsize\textwidth\hsize
\columnsep.5cm
\advance\textwidth\hsize
\advance\textwidth\columnsep
\columnwidth\hsize
\count0=1
%
\def\rtitle{\runtit}
\def\issue{MAPS 94.1}
\let\ftn\footnote
\catcode\@=11
\def\@oddhead{\it\issue\hfil\rtitle}
\def\@evenhead{\it\rtitle\hfil\issue}
\def\@oddfoot{\rlap{Draft \today}\hfil
-\thepage-\hfil\llap{\copyright cgl}}
\def\@evenfoot{\rlap{\copyright cgl}\hfil
-\thepage-\hfil\llap{Draft \month/\year}}
\catcode\@=12
\endinput %cgl@risc1.rug.nl
```

B.3 Customing: tugproc.cus

```
%tugproc.cus Dec 93
\hoffset-.75cm\overfullrule0pt
\normalcollgt24.5cm\collgt\normalcollgt
\pagewd17.5cm
\intercolwd.5cm
\resetpagelgt\twocol \pageno1
\tubpagelgt\pagelgt%Weird
\enablemetacode
\everyverbatim{\enablemetacode}
%
\def\rtitle{\hbox to \pagewd{\tenpoint
\issue\hfill{\it\runtit}}}
%
```

```
\def\rfoot{\hbox to \pagewd{\tenpoint
\rlap{Draft \today}\hfill-\folio-\hfill
\llap{\copyright cgl}}}
%
\def\issue{MAPS 94.1}
%with
\let\small\sevenrm \let\tiny\fiverm
\def\quote{\endgraf\bgroup\narrower
\smallskip\noindent}
\def\endquote{\smallskip\egroup\endgraf
\noindent}
\newcount\fcnt
\def\ftn{\advance\fcnt1
\footnote{$\{ \}^{\the\fcnt}$}}
\endinput %cgl@risc1.rug.nl
```

B.4 Customing: ltugproc.cus

```
%ltugproc.cus Dec 93
\hoffset-.5cm\voffset-.25cm
\overfullrule0pt
\hsize=8.75cm \vsize=25cm
\columnsep=.5cm
\textheight\vsize \textwidth\hsize
\advance\textwidth\hsize
\advance\textwidth\columnsep
\pagewd\textwidth
\columnwidth\hsize
%
\catcode\@=11
\def\@oddhead{\small
\issue\hfill{\it \runtit}}
\let\@evenhead\@oddhead
\def\@oddfoot{\tiny \rlap{Draft \today}
\hfil-\thepage-\hfil
\llap{\copyright cgl}}
\let\@evenfoot\@oddfoot
\catcode\@=12
%
\setcounter{page}{1} %\count0=1
\def\issue{MAPS 93.1}
\let\ftn\footnote
\endinput %cgl@risc1.rug.nl
```

B.5 Customing: eurotex.cus

```
%eurotex.cus Dec 93
\hoffset-.75cm \voffset-.25cm
\hsize=18cm \vsize=25cm
\textwidth\hsize \textheight\vsize
\columnwidth\hsize \linewidth\hsize
\topmargin0cm \overfullrule0pt
\evensidemargin1cm \oddsidemargin1cm
\let\ftn\footnote
%
\catcode\@=11
\@colht\vsize \@colroom\vsize%Note!?!
\def\@evenfoot{\rlap{Draft \today}\rm\hfill
-\thepage-\hfill\llap{\copyright cgl}}
\let\@oddfoot\@evenfoot
\catcode\@=12
\endinput %cgl@risc1.rug.nl
```

Appendix C: tug.ppt

The idea is to propose a preprint style which shields authors from the differences in markup in for example `tugboat.sty`, and `tugproc.sty`. It is in the same spirit as the preprint style of the AMS, and the one of Elseviers Science Publishers announced of late. It is not equivalent to `tugboat.sty`

- no options
- no spanning-columns

- various inner macros have been split-off⁷⁵

For me it is an efficient replacement of `tugboat.sty`, provided I include my macros for

- verbatim mode suite
- bordered table
- bibliography handling, and
- some special macros required by the subject.

If this style—or at least the idea—is adopted by LUGs too, then authors can mark up their copy by this style and submit it to any LUG bulletin, without change of markup. Sounds like a nice and cooperative idea to me.

Whatever the value of `tug.ppt`, it is undoubtedly useful for educational purposes, in the sense that it shows what `tugboat.sty` is essentially all about, functionally. My case rest.

```

1. %Adapted from TUGboat.sty, essentially
2. %2-column \xcol is the column number
3. %within a page; ranges from 1 to \maxcols
4. \newcount\xcol
5. \newcount\maxcols
6. %
7. \newdimen\pagewd
8. \newdimen\colwd
9. \newdimen\intercolwd
10. %
11. \catcode'\@=11
12. % remove \outer
13. \def\newbox{\alloc@4\box\chardef\insec@unt}
14. \def\boxcs#1{\box\csmname#1\endcsname}
15. \def\setboxcs#1{\setbox\csmname#1\endcsname}
16. \def\newboxcs#1{\expandafter
17.   \newbox\csmname#1\endcsname}
18. \newboxcs{column1}
19. \newboxcs{column2}
20. %
21. \def\midpage#1{\vbox{\ifnum\xcol=\maxcols
22.   \runheadelse\null\vskip\baselineskip\fi
23.   \kern2ex
24.   \vbox to\vsize{%
25.     \hbox to\pagewd{#1}\vss}
26.   \kernlex
27.   \ifnum\xcol=\maxcols
28.     \runfootelse\vskip4ex\fi}}
29. %
30. \def\runhead{\hbox to\pagewd{\sevenrm
31.   \issue\hfill{\it\runtit}}}}
32. %
33. \def\runfoot{\hbox to\pagewd{\fiverm
34.   \rlap{Draft \today}\hfill--\thepage
35.   --\hfill\llap{\copyright cgl}}}}
36. \def\thepage{\today}
37. \def\today{\ifcase\month\or Jan\or Feb\or
38.   March\or April\or May\or June\or July\or
39.   Aug\or Sept\or Oct\or Nov\or
40.   Dec\fi\space\number\day, \number\year}
41. %
42. \def\newcol{\endgraf\vfill\eject}
43. %
44. \def\newpage{\vfill\eject
45.   \loop
46.   \ifnum\xcol>1
47.     {\leavevmode\endgraf\vfill\eject}
48.   %\xcol is advanced in the output routine
49.   \repeat}
50. % horizontal offset of column
51. % from left edge of page
52. \newdimen\coloffset \coloffset=\z@
53. \def\incrcoloffset{%
54.   \global\advance\coloffset\colwd
55.   \global\advance\coloffset\intercolwd}
56. %
57. \output={\global\setboxcs{column\number\xcol}=
58.   \midpage{\kern\coloffset\pagebody\hfil}
59.   \incrcoloffset
60.   \ifnum\xcol=\maxcols
61.     \shipout\hbox{\global\xcol=@ne
62.       \loop\rlap{\boxcs{column\number\xcol}}%
63.       \ifnum\xcol<\maxcols
64.         \global\advance\xcol@ne
65.         \repeat
66.       \hbox to\pagewd{\hss}}%
67.     \global\advance\count0@ne
68.     \global\coloffset\z@
69.     \global\xcol=@ne
70.   \else
71.     \global\advance\xcol@ne
72.   \fi}
73. \catcode'\@=12
74. %
75. \def\title*#1*{\def\thetitle{#1}}
76. \def\author*#1*{\def\theauthor{#1}}
77. \def\address*#1*{\def\theaddress{#1}}
78. \def\netaddress*#1*{\def\thenetaddress{#1}}
79. \def\article{\hrule\kern2ex\noindent
80.   {\bf \thetitle}\medskip\noindent
81.   \item{\theauthor}
82. \def\endarticle{\makesignature
83.   \global\xcol\maxcols\vfil\eject\end}
84. \def\head*#1*{\goodbreak\bigskip\noindent
85.   {\bf #1}\medskip\noindent\ignorespaces}
86. \def\subhead*#1*{\medskip\noindent{\bf #1}}
87. \def\subsubhead*#1*{\smallskip{\bf #1}}
88. \def\tubissue#1(#2){\TVB~#1, no.~#2}
89. \def\\{\hfil\break}
90. \def\makesignature{\medskip
91.   \rightline{\hbox to.5\hsize{\strut
92.     \llap{\$diamond\$quad}\theauthor\hss}}
93.   \rightline{\vbox{\noindent
94.     \hsize=.5\hsize
95.     \theaddress\endgraf\noindent
96.     \thenetaddress}}}}
97. %
98. % Defaults
99. %
100. \def\onecol{\maxcols=1
101.   \hsize=16cm
102.   \pagewd=\hsize
103.   \colwd=\hsize
104.   \vsize=25cm
105.   \maxcols=1
106.   \xcol=1
107. }
108. %
109. \def\twocol{\maxcols=2
110.   \hsize=8.75cm
111.   \colwd=\hsize
112.   \intercolwd=.5cm
113.   \pagewd=18cm
114.   \vsize=25cm
115.   \maxcols=2
116.   \xcol=1
117. }
118. \hoffset-1cm
119. \voffset-1cm
120. \twocol
121. \endinput
Contents
- OTR

```

⁷⁵Also useful in other contexts than *TUGboat*.

```

\boxcs.....14
\setboxcs.....15
\newboxcs.....16
\midpage..... 21-28
\runhead.....30-31
\runfoot.....33-35
\thepage.....36
\today.....37-40
\newcol.....42
\newpage.....44-49
\incrcoloffset.....53-55
\output.....57-72
- Markup structures
  \title.....75
  \author.....76
  \address.....77
  \netaddress.....78
  \article.....79-81
  \endarticle.....82-83
  \head.....84-85
  \subhead.....86
  \subsubhead.....87
- Miscellaneous
  \tubissue.....88
  \.....89
  \makesignature.....90-96
- Defaults
  \onecol.....100-106
  \twocol.....109-117
- Initializations      118-120
%
%Split-off---inner level---are
% - tugboat.cmn abbreviations
% - size switching macros
% - \quote...\endquote
% - \bitem (essentially plain's \item)
% - verbatim mode macros
% - (bibliography macros,
%   \btable...\endbtable,
%   ...)
%Author: C.G van der Laan, Hunzeweg 57,
%        9893PB Garnwerd, The Netherlands,
%        05941-1525 cgl@risc1.rug.nl
%History of changes
%March 94   Essential set up

```

I have used the `tug.ppt` style already for my articles BLUE's Bibliography, Transparencies and Verbatim.

A bare-to-the-bones sample is

```

%test tug.ppt      %cgl@risc1.rug.nl
\input tug.ppt
\input cgl.mac     %My inner level macros
%\input tug.fonts %Size switching macros
%\input tug.abr   %Abbreviations and logos
%
\def\runtit{tug.ppt}
\title*\runtit---{\rm the alternative}*
\author*Kees van der Laan*
\address*Hunzeweg 57, 9893PB\
        Garnwerd,
        The Netherlands*
\netaddress*cgl@risc1.rug.nl*
\def\issue{MAPS 94.1}
\article
\keywords ppt, education.\endkeywords
\abstract A preprint style for TUG publications
\endabstract
%
\head*Why?*
Although it seems misplaced at first
sight---apart from tugboat.sty and
tugproc.sty---it can be worthwhile for
the \TeX{} community at large to provide
for a preprint style, such that

```

each author can submit a paper in preprint style, while the editorial board at hand substitutes the preprint style by the style needed for the concrete bulletin. This approach is similar to AMS' approach.

```
\endarticle
```

Appendix D: Contents

The structure of this article is a bit complex because of the modular treatment of the various style files. The article begins with a general part followed by specific parts, each devoted to a style file. At the end there is again a general part which contains 'Looking back' and the like. The appendices reflect the same splitting in parts, where each part is again devoted to a specific style file.

Abstract

Introduction

Warnings!

- Generic coding pitfall
- Why?
- What?
- Functionalities
- Developers
- Notations

What does *TUGboat* look like?

TUGboat processing

- An alternative approach?
- Annoyance
- Refereeing*

Design

Coding conventions

What is provided by the styles?

- Contents of the style files
 - tugboat.cmn
 - tugboat.sty
 - ltugboat.sty
- One-ness
- For editors only

Conclusion

What does the TUG AM proceedings look like?

Processing proceedings TUG annual meetings

- Refereeing
- TUG proceedings styles

tugboat.sty

Customing

- Page size
- Headers and footers

Coding, or T_EXies at work

- Fonts
- Conventions
- Outer from tags
 - General mechanism for tags
- Checking ahead
- DeleteOptionalSpacesandPars

- Options
 - Sectioning commands
 - Lists
 - Alternative
 - Intermezzo: prefix versus postfix
 - What confuses me
 - Verbatims
 - How are verbatims processed?
 - Alternative: parameterization over end delimiter
 - Intermezzo: functionalities
 - Knuth's verbatims
 - Intermezzo: block comment
 - Alternative: the big deal
 - In-line versus display
 - Options via `\thisverbatim`
 - Coding alternative
 - Alias tags
 - Placement of figures
 - Argument processing on the fly
 - `\if@savingargument`
 - Page make-up, or the OTR
- Conclusion

- Customizing
- Page size
 - Headers and footers
 - Sectioning commands
 - Footnotes
 - Lists
 - Verbatims
 - Figures
 - Bibliography
- Conclusion

- Customizing
- Page size
 - Headers and footers
 - Abstract
 - Bibliography
 - Appendices
 - Coding: alternative
 - Switching into `tugboat.sty`

Conclusion

- Customizing
- The page size
 - The headers and footers
 - Section heads
 - Appendices
 - Bibliography

Conclusion

What does the Euro \TeX proceedings look like?

- Customizing
- Page size
 - Headers and footers
 - Intermezzo: `def` versus `toks`

Conclusion

- Customizing
- Page size
 - Headers and footers
 - Title
- Conclusion

Looking back

- \TeX ing paradigms

Looking forward

Acknowledgements

Bibliography

Appendices

- A Templates *as is*: `<style>.tem`
- B Customizing: `<style>.cus`
- C `tug.ppt`
 - contents `tug.ppt`
 - sample
- D Contents

Book review

Nico Poppelier

n.poppelier@elsevier.nl

Abstract

Two new books are discussed: *Practical SGML* from Eric Herwijnen, and ‘math into T_EX, a simple introduction to $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX’ from George Grätzer.

Practical SGML

Eric van Herwijnen. Kluwer 1994, 284 pages (including indexes). To be published.

In my review of the first edition of *Practical SGML* by Eric van Herwijnen (TUGboat 13, #2), I praised it as ‘one of the best books on SGML currently available.’ It still is one of the few books on practical application of SGML, by someone who has used SGML in practice rather extensively. The new edition has undergone significant changes with respect to the previous one. Unfortunately, they are not all changes for the good: the book still contains a lot of practical information – more than the first edition – but is not a *better book*.

As a reference work the quality of the book has certainly improved. More material has been added, and the book has been largely re-structured. The previous edition consisted of three parts, *Getting started with SGML*, *Advanced SGML* and *SGML implementations*. The new edition has more chapters, grouped together in four parts, *Getting started*, *Writing a DTD*, *Customizing SGML* and *Special applications*. Especially the second part, about how to write a dtd (document type definition), has improved a lot, with chapters on document analysis, structure diagrams, and the various declarations one can find in a dtd. Part III, about customizing SGML, describes the SGML declaration, and SGML features such as minimization, marked section and short references. It also describes the problems that can arise with ambiguous definitions, and gives advice about how to avoid ambiguities. Under the heading of ‘Special applications’ (part IV) Mr. van Herwijnen discusses SGML and EDI, SGML and mathematics, and SGML and graphics. He also explains the relation between SGML and other ISO standards, such as, e.g., DSSSL and SPDL. In all the examples in the book the public-domain `sgmls` parser is used, which makes it possible for most readers to try the examples for themselves.

On the negative side however: so much material is now contained in the book, especially in the form of figures and tables, that the book, in my opinion, is not a *pleasant-to-read* introduction to SGML anymore. Another thing which I find rather distressing, at least in the pre-publication copy the author kindly sent me, is the design: the book uses too

many fonts, in sometimes unharmonious combinations, the distribution of vertical space is uneven, and the placement of tables and figures leaves a lot to be desired. A possible explanation could be that this new edition of *Practical SGML* was prepared using SGML, and was formatted using Adept 5.0 from ArborText Inc. Obviously, designing a book that is comfortable to read is not the same as writing a ‘FOSI’, an output specification for ArborText’s Adept product. I hope that the publisher will work hard on improving the layout of the book, but I have my doubts.

Of course, this says nothing about the applicability of SGML to book production, but only about the quality of available SGML tools, or the expertise of the people using these tools. That computers *are* capable of producing more readable and more attractive books is shown by a book co-authored by one of Mr. van Herwijnen’s colleagues at CERN, namely *A L^AT_EX Companion*, by Michel Goossens, Alexander Samarin and Frank Mittelbach. But then, of course, that book was made with L^AT_EX!

Math into T_EX, A simple introduction to $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX

George Grätzer. Birkhäuser 1993, 294 pages (including indexes), ISBN 0-8176-3637-4. (Includes diskette).

Even though document-preparation packages of the *wysi-wyg* type become better and better every year, the majority of mathematicians, computer scientists and physicists still use T_EX, in any of its many flavours. In 1982 the American Mathematical Society released $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX, which has certainly contributed to the popularity of T_EX in mathematician’s circles. Around 1990, the macros of $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX were made available for the expanding community of L^AT_EX users in the form of $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX. And now, finally, there is a book that describes this useful but complex extension to L^AT_EX for beginners.

George Grätzer teaches mathematics at the University of Manitoba (Canada), and has tried to write a book ‘from a user’s point of view’. His book consists of three parts. In part I, *A short course*, he explains how to install $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX on a IBM-compatible pc under DOS, and on an Apple Macintosh, and then explains the basics of $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX. In part II, *A leisurely course*, he goes back to

the fundamentals of typing text and formulas in $\text{T}_{\text{E}}\text{X}$, and then explains all a mathematician, or engineer or scientist, needs to know about $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Finally, part III is about customizing.

Math into $\text{T}_{\text{E}}\text{X}$ was written with the basic idea behind the design of $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ nestled firmly in the mind of the author. Mr. Grätzer emphasizes proper usage of $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ while writing papers and books. His approach is didactically very good, he takes his time explaining things, and gives enough examples. *Math into $\text{T}_{\text{E}}\text{X}$* is not a book about $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ itself, so the author does not cover all of $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Nevertheless, he treats tables (briefly) and $\text{Bib}_{\text{T}_{\text{E}}\text{X}}$, which makes the book a useful introductory text as well as a handy reference for

authors who do not want to know more about $\text{T}_{\text{E}}\text{X}$ and its flavours than is necessary for writing a research paper.

Even though many people, including several well versed in $\text{T}_{\text{E}}\text{X}$ and $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, read draft versions of *Math into $\text{T}_{\text{E}}\text{X}$* , the book contains a lot of errors. One of the more interesting ones is this one: in section 3-4.8, on hyphenation, the author gives the following example

```
\hyphenation{data-base Birkh-h\"auser}
```

I often wish that $\text{T}_{\text{E}}\text{X}$ could do tricks like this, but alas!

These errors will be removed in a second, revised edition I assume (if enough people buy the book), and then I can say in all truth that *Math into $\text{T}_{\text{E}}\text{X}$* is a welcome addition to the growing collection of books about $\text{T}_{\text{E}}\text{X}$ and related topics.

The T_EX Hierarchy

Donald Arseneau^{*}, Raymond Chen[†], Victor Eijkhout[‡]

Introduction

For the UNIX operating system, a list of characterizations exists describing what constitutes a novice, a user, a guru, . . . Here we give a similar list for users of T_EX. The reader is kindly asked to take this purely in a humorous vein.

The name

Novice says ‘tecks’.

User says ‘tecchhh’ but still moistens the screen doing it.

Programmer correctly pronounces ‘T_EX’.

Wizard has made at least one bad pun on the name T_EX.

Guru knows that even Knuth says ‘tek’.

The manual

Novice owns the TeXbook.

Programmer has just made a first correction to the text.

Hacker has formatted `texbook.tex` and knows about Knuth’s ‘little joke’.

Wizard is thinking of ways to supply the missing ‘tactile and olfactory sensations’ of T_EX.

Guru thinks ‘T_EX: the program’ is more useful.

The index of the T_EX book

Novice is confused by the number of references for each entry, has laughed at ‘T_EX: bad puns on the name’, and has counted the number of middle names of Barbara Beeton.

User knows about Bo Derek (in the T_EXbook), Jill Knuth (in the Metafont book), and Ellen Gilkerson (in the L^AT_EX book).

Wizard knows why some entries are italicized or underlined.

Guru knows to look up Bourbaki for smart line breaks in paragraphs.

The system

Novice has found many bugs in T_EX.

User has learned that there are no bugs in T_EX, but doesn’t understand why ‘it doesn’t work!’.

Guru has actually found bugs in T_EX; frames the check from DEK.

Guru extraordinaire cashes checks from DEK.

Famous people

Novice is not sure whether Leslie Lamport is a man or a woman.

User knows not to capitalize ‘barbara beeton’.

Wizard knows how to pronounce ‘Knuth’ and ‘Eijkhout’.

Guru Knuth has asked about their middle name(s).

Programming style

Novice uses grouping without knowing why.

User writes `\bf{ . . . }` and doesn’t understand what went wrong.

Programmer writes `\def\bold#1{\bf #1}`.

Programmer first class writes `\long\def\bold#1{\bf #1}`.

Hacker writes `\def\beginbold{\bgroup\bf}`
`\def\endbold{\egroup}`

Wizard writes `\def\bold{\bgroup\bf\let\next=}`.

Guru writes `\def\bold#{\bgroup\bf\let\next=}`.

Style (cont’d)

Novice has heard of ties.

User inserts ties and writes ‘`dr.\ ’`’.

Hacker writes ‘`dr.\ ’`’, except in bibliographies where `frenchspacing` is in effect.

Guru Makes ‘.’ an active character in bibliographies so that ‘D.E. Knuth’ means ‘`D.\,E.\penalty 300\ Knuth`’.

Errors

Pre-novice wonders why ‘Q’ takes so long to quit.

Novice will exit on the first ‘error’, even if the message starts with ‘OK’.

User keeps pressing return to scroll past errors, until that gets into an infinite loop.

Guru having written the input file with ‘`cat >`’ in the first place, the guru will type ‘i’ at an error, correcting all typos and supplying all missing mac-

^{*}Triumpf, 4004 Wesbrook Mall, Vancouver BC, Canada V6T 1Zt, asnd@erich.triumf.ca

[†]Microsoft Corporation, One Microsoft Way, Redmond WA 98052-6399, raymond@microsoft.com

[‡]Department of Computer Science, University of Tennessee at Knoxville, Knoxville TN 37996-1301, eijkhout@cs.utk.edu

ros interactively, thereby successfully completing the formatting in the first run.

Capacity Exceeded . . .

Novice constantly runs into the ‘TeX capacity exceeded’ error and asks the admin to build a larger version.

User knows how to find unbalanced curly braces.

Hacker Occasionally runs into the ‘TeX capacity exceeded’ error and usually finds a way around them.

Wizard knows how to increase T_EX’s capacity, taking care to read DEK’s warnings about setting the values too high.

Guru ignores DEK’s warnings.

Printing and previewing

Novice prints the whole document after each run of T_EX.

User knows of previewers.

Programmer knows at least two previewers and vigorously argues why one is utter garbage.

Wizard thinks that

```
\tracingoutput=1
\showboxdepth\maxdimen
\showboxbreadth\maxdimen
```

is the best previewer.

Macros

Novice has heard of macros, but has never seen one.

User writes macros that are used once, and that are longer than the code they replace.

Programmer having been bitten by unwanted spaces, writes macros that don’t contain spaces, and every line ends with a ‘%’.

Hacker has written self-modifying macros, writes `\endlinechar=-1` or `\catcode\^M=9` to prevent having to put %’s at the end of lines in macros.

Guru has written macros containing #####, more than 3 `\expandafter`’s in a row, and the sequence `\expandafter\endcsname.item[Fossil]` Still has macros written in T_EX78.

Macros (cont’d)

Novice has written a macro `\box` to draw a box.

User has renamed it to `\boxit`.

Wizard has redefined `\mbox` so that it can have `\verb` in its arguments.

L^AT_EX

Novice uses L^AT_EX because colleagues and friends do.

User uses L^AT_EX, even though colleagues and friends use Microsoft Word, or Word Perfect.

Wizard uses L^AT_EX for journal and conference submissions, but homegrown macros when working alone.

L^AT_EX errors

Novice actually takes the manual when it says ‘LaTeX error. See LaTeX manual for explanation.’

User knows what the relevant bits of L^AT_EX error messages are.

Programmer knows what to type at the question mark when L^AT_EX reports ‘`\begin{document}` ended by `\end{itemize}`’.

Wizard doesn’t make errors in L^AT_EX, and answers questions about L^AT_EX by editing `latex.tex`.

Guru knows whether to edit `latex.tex`, `lplain.tex`, `article.sty`, or `art10.sty`.

L^AT_EX style

Novice types `a_{1}` because the error in `a_{1}` occurred on the ‘_’.

User types `$a_{1}` because Leslie Lamport says so.

Other packages

Novice could do more in Pagemaker.

User doesn’t see the difference between T_EX macros and Word Perfect macros.

Hacker writes macros to make T_EX look more like troff.

Wizard types `\input troff` to process old troff files.

Guru types

```
\input txtmacros
\input text.txt
```

to format plain text.

Life, everything

Novice thinks that learning T_EX will take a long time.

User realises that it wasn’t so bad after all.

Programmer tries to convince himself that the next macro is really going to save time in the future.

Wizard daydreams idly about how much he could have done with his life if he had never heard of T_EX.

Guru realises that a life without T_EX is not worth living.

(Also thanks to Barbara Beeton, Tim Chow, Denys Duchier, Dan Ellard, Michael Sofka.)

Obtaining T_EX*

Sebastian Rahtz

ArchaeoInformatica, 12 Cygnet Street,
York YO2 1AG, England
spqr@ftp.tex.ac.uk

January 1994

From the network

The UK T_EX Archive (Internet ‘Daughter’ archive) on `ftp.tex.ac.uk` is part of a collaborating network of archives organised by the T_EX Users Group known as CTAN (Comprehensive T_EX Archive Network). The three main archives now follow the same structure and have identical files (`ftp.tex.ac.uk`, `ftp.shsu.edu` and `ftp.uni-stuttgart.de`).

The preferred access method to the UK T_EX Archive is using the *gopher* program which has a set of useful indexes to help you locate what you are looking for, but Internet *ftp* access is also very common. JANET users may only access the machine using the `ft-relay` site, as it has no X25 connection.

The CTAN archives all run an enhanced *ftp* server which supports dynamic compression, uncompression, and archive creation options. Fetch the top-level file `README.archive-features` for information. The server also supports site-defined commands to assist you. Please read `README.site-commands` for a brief overview.

The main directories which make up CTAN are listed below; readers are referred to David Jones’ *Index of T_EX Styles and Macros* for details of macro packages and individual style files. This can be found in CTAN as `info/tex-styles-and-macros.txt`

biblio contains bibliography-related files, such as BibT_EX.
digests contains back issues of T_EX-related periodicals. various aspect of T_EX.
dviware contains the various dvi-to-whatever filters and drivers.

fonts contains a collection of fonts, both sources and pre-compiled.
graphics contains utilities and macros related to graphics.
help contains files which provide an overview to the archive and the T_EX system.
info contains files and tutorials which document
indexing contains utilities and related files for indexing documents.
language contains non-English related implementations of T_EX.
macros contains macros for T_EX and its derivatives in unique subdirectories.
support contains files and programs which can be used in support of T_EX.
systems contains complete system setups, organized by operating system.
tools contains the various archiving tools which users may find useful.
web contains WEB-related files and utilities.

Details of where to find public domain, or shareware, T_EX packages for different machines and operating systems are given in Table 1. Some readers may prefer to purchase a commercial package, with support. *Neither CTAN nor UK-TUG are in a position to offer support for software in the archive.*

Please report any problems with CTAN archives via e-mail to `ctan-mgr@shsu.edu`. The entire archive will be available on CDROM from spring 1994 from Prime Time Freeware, 370 Altair Way, Suite 150, Sunnyvale, CA 94086 USA (`ptf@cfcl.com`).

*Reprint from the Annals of the UK T_EX Users’ Group: **Baskerville**, Volume 4.1, Januari 1994.

Name	Environment	CTAN path	Notes
emT _E X	DOS, OS/2	systems/msdos/emtex	the betatest drivers, and 386-specific versions of T _E X and MF, are in the betatest subdirectory
sbT _E X	DOS	systems/msdos/sbtex	includes latest T _E X and Metafont
gT _E X	DOS	systems/msdos/gtex	386-only T _E X and Metafont, which replace parts of emT _E X, and work with Windows memory management
T _E Xas	DOS	systems/msdos/texas	Large T _E X, which replaces T _E X386 in emT _E X, and works with Windows memory management
OzT _E X	Mac	systems/mac/oztex	This package is now shareware; there is also a multi-lingual version (Euro-OzT _E X)
CMacT _E X	Mac	systems/mac/cmactex	Port of Unix T _E X, including Rokicki's dvips. A fuller version (with large memory) can be purchased.
DirectT _E X	Mac	systems/mac/directtex	
pasT _E X	Amiga	systems/amiga/pastex	
—	Atari	systems/atari/lindner-tex-packed-disks	
—	Atari	systems/atari/cs-tex	
web2c T _E X	Unix	systems/unix/web2c	a complete source kit for T _E X and Metafont, which should compile on most Unix boxes; needs a C compiler.
sparctex	Sun Sparc	systems/unix/unixkit	
—	DEC Alpha	systems/unix/alpha	for Alpha running Unix
—	Xenix	systems/unix/xenix	
Decus T _E X	VAX/VMS	systems/vms/decus	
—	Alpha VMS	systems/vms/alpha	for Alpha running VMS

Table 1: CTAN location of T_EX systems for different machines

Unix tapes

For a complete Unix T_EX distribution, a 1/4 inch cartridge, QIC-120 or QIC-150 format (DC600A or DC6150) can sent with envelope *and* stamps for return postage to:

David Osborne
Cripps Computing Centre,
University of Nottingham,
Nottingham NG7 2RD

Due to currency exchange, this service is offered only within the UK.

The international T_EX Users Group can also supply many T_EX materials on disk. Contact:

T_EX Users Group
PO Box 869
Santa Barbara, CA 93102
USA
Phone: 805-899-4673 E-mail: tug@tug.org

L^AT_EX_{2 ϵ} — Second Test Release Available

Leslie Lamport and the L^AT_EX₃ project team

14 February 1994

The second test release of L^AT_EX is now available. Notes about this release are at the end of this announcement. The first release revealed a few serious bugs and several other areas which need attention. Many people reported that documents and style files had been processed by the new system without serious problems; this includes one very important test — the original text of ‘The L^AT_EX Manual’! We can also report that this release has been tested with all the files in the L^AT_EX 2.09 test suite and produces only predictable differences in the output.

First, here is the original announcement

L^AT_EX_{2 ϵ} is the new standard version of L^AT_EX — prepared and supported by the L^AT_EX₃ project team. It is upwardly compatible with L^AT_EX 2.09 documents, but contains new features.

Over the years many extensions of L^AT_EX have been developed. This is, of course, a welcome development, since it shows that the L^AT_EX system is in a healthy state. It has, however, had one unfortunate consequence: there are now several incompatible systems, in the sense of format (.fmt) files, all claiming to be L^AT_EX. Therefore, in order to process documents coming from various places, a site maintainer needs to provide several format files: L^AT_EX (with and without NFSS), S_Li_TE_X, $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX, and so on. In addition, when looking at a source file it is not always clear for which format the document was written.

L^AT_EX_{2 ϵ} puts an end to this unsatisfactory situation — it will give access to all such extensions based on a single format and thus end the proliferation of mutually incompatible dialects of L^AT_EX 2.09.

It uses an enhanced version (NFSS2) of the New Font Selection Scheme. Files such as `amstex.sty` (formerly the $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX format) or `slides.sty` (formerly the S_Li_TE_X format) will become extension packages, all working with this single format.

The introduction of this new version will also make it possible to add a small number of often-requested features (such as extended versions of `\newcommand`).

To summarize:

- **Standardisation:**
a single format incorporating NFSS2, to replace the present multiplicity of incompatible formats — NFSS, `lfonts`, `pslfonts`, etc.

- **Maintenance:**

a standardised system supported by a reliable maintenance policy.

L^AT_EX_{2 ϵ} adheres, as far as possible, to the following principles:

- Unmodified version 2.09 document files can be processed with L^AT_EX_{2 ϵ} .
- All new features of L^AT_EX_{2 ϵ} conform to the conventions of version 2.09, making it as easy as possible for current users to learn to use them.

L^AT_EX_{2 ϵ} is described in a new edition of ‘L^AT_EX: A Document Preparation System’ by Leslie Lamport (to appear during 1994) and ‘The L^AT_EX Companion’ by Goossens, Mittelbach and Samarin, both published by Addison-Wesley.

L^AT_EX_{2 ϵ} will be distributed twice a year. This distribution is a preliminary test release, and doesn’t contain all of the files that will be part of the full release. In particular, it does not contain the planned extensions in the area of graphics inclusion.

The first full release will be available in Spring 1994.

This is a test release, so please get it and test it on as many different systems as possible!

Getting the second test release

L^AT_EX_{2 ϵ} can be retrieved by anonymous ftp from the CTAN archives:

```
ftp.tex.ac.uk    /tex-archive/macros/latex2e/core
ftp.shsu.edu    /tex-archive/macros/latex2e/core
ftp.dante.de    /tex-archive/macros/latex2e/core
```

Bug reports

Please report any problems with L^AT_EX_{2 ϵ} by using the report-generating program `latexbug.tex`, included in the L^AT_EX_{2 ϵ} distribution. Error reports should be sent to the following e-mail address:

`latex-bugs@rus.uni-stuttgart.de`

Help for style/package/class file maintainers

We shall be producing a short document summarising the changes that may be needed in order to turn a style file for old L^AT_EX into a class or package file. To get the latest information on this subject or to report problems, please send a message to the following address:

`bugs@minnie.ZDV.Uni-Mainz.de`

Please ensure that it starts with the following line *exactly*:

```
>Category: latex-class-writing
```

If you want a prompt and useful reply, please make your message as short and precise as possible.

Notes on the second test release

- Only the straightforward bugs and the serious bugs found in the first release have been fixed; other problems are still being worked on. Some of the reports uncovered features of the original LaTeX which some consider to be bugs; whether these should be ‘fixed’ is not always clear.
- There are thus several areas, including things arising from bug reports, on which we are still working.
- The details of the distribution, unpacking etc are still under development.
- The details of the typography in the standard classes is not finalised.

- The precise functionality of the compatibility mode is not fixed.
- Still no graphics package!

Warning

If you have installed the first release then you must remove it completely and start again. This is because there are many small differences, including new file names, and because the installation procedure now behaves differently.

For the L^AT_EX₃ Project

Johannes Braams
David Carlisle
Alan Jeffrey
Frank Mittelbach
Chris Rowley
Rainer Schöpf

Literate Programming FAQ

version 1.0.1

David B. Thomas

15 october 93

Welcome to the Literate Programming Frequently Asked Questions List

This version was created Fri 10/15/1993 and should be considered stale after 90 days. Information contained in this document is the best available at preparation.

Disclaimer: ‘This FAQ is presented with no warranties or guarantees of ANY KIND including correctness or fitness for any particular purpose. The author of this document has attempted to verify correctness of the data contained herein; however, slip-ups can and do happen. If you use this data, you do so at your own risk.’

Copyright 1993 David B. Thompson. All rights reserved. Permission is granted to copy this document for free distribution so long as it remains intact and unmodified.

1 Introduction

This document is for new and experienced users of literate programming tools. The purpose is to explain the concept of literate programming and to provide a resource for locating files of interest to literate programmers and those interested in literate programming.

The Literate Programming (LitProg) Frequently Asked Questions (FAQ) list is maintained by Dave Thompson, who can be reached at:

thompson@sun1.coe.ttu.edu	Preferred mailing address for FAQ related comments/questions.
dthompson@coe2.coe.ttu.edu	Preferred for purposes other than FAQ.
wqdbt@ttacs1.ttu.edu	Forwarded to me.

Comment and constructive criticism is welcome. Direct flames to /dev/null (or > nul if you're a msdos user!) If you find an error, please report it. I'm particularly interested in establishing the locations of generally available literate programming tools. If you are the author of such a tool and wish to have it included in this list, please send email.

Please note this is a work-in-progress. It is *not* complete, and probably will not be complete for some months. Nevertheless, the information contained herein may be useful to some. Use it as it is intended.

2 Table of Contents

- Introduction
 - Typography.
- How to get the FAQ.
 - Literate Programming FAQ.
 - FWEB FAQ.

- The LitProg newsgroup.
- Internet nodes of interest to literate programmers.
- What is literate programming?
- How do I begin literate programming?
- What tools are available and where are they?
 - FWEB
 - CWEB
 - NOWEB
 - NUWEB
 - FunnelWeb
 - SchemeWEB
 - c2latex
- Languages supported with literate programming tools.
- What other resources are available?
- Code examples.
 - Cameron Smith's KR-CWEB
 - Bart Child's examples for FWEB
 - Examples with CWEB
 - Examples with FWEB
- Bibliographies.
- How to anonymously ftp.
- Acknowledgements.
- Notes.

3 How to get the FAQ

3.1 Literate Programming FAQ

You have many ways to get a current copy of this FAQ. One is to use anonymous ftp (if you don't know how, see a later section in this FAQ) to connect to one of the Comprehensive TeX Archive Network (CTAN) sites or the literate programming archive and retrieve a copy of the file. (For more information on CTAN and the literate programming archive, see the section below entitled ‘Internet Nodes of Interest to Literate Programmers.’)

An alternative is to use the fileserver at SHSU. Send a message to FILESERV@SHSU.EDU (FILESERV@SHSU.BITNET) and include in your message

```
SENDME FAQ.LITPROG
```

The file server will forward a copy of the file to you via email.

3.2 FWEB FAQ

Marcus Spoh marcus@x4u.desy.de maintains the FWEB FAQ. It can be retrieved in the same way as this FAQ; either by anonymous ftp or through the SHSU file

server. On the SHSU server, the file name is FAQ.FWEB. Invoke your ftp software, open a connection to NIORD.SHSU.EDU [192.92.115.8], attach to the directory FAQ, and transfer the file FAQ.FWEB.

Alternatively, send a message to the file server, FILESERV@SHSU.EDU (FILESERV@SHSU.BITNET), and include the following text in a one line message:

```
SENDME FAQ.FWEB
```

The file server will send the current version of the file via email.

The FWEB FAQ exists in various formats, including HyperText (see other resources below). In Europe, the complete distribution can also be obtained from ftp.desy.de [131.169.10.38] in directory /pub/faq/web/fweb/.

4 The LitProg Newsgroup

One of the most important resources is the literate programming newsgroup, comp.programming.literate. You can read this newsgroup using your standard reader. Alternatively, the newsgroup is gated to a mailing list hosted by George Greenwade and Sam Houston State University. You can subscribe by sending mail to the list-server, LISTSERV@SHSU.EDU (or LISTSERV@SHSU.BITNET), and include in the message one line of text:

```
SUBSCRIBE LITPROG 'your name in quotes'
```

The list is unmoderated; messages sent to litprog@shsu.edu are automatically distributed to all subscribers and cross-posted to comp.programming.literate.

Archives of the mailing list and newsgroup are maintained on niord.shsu.edu [192.92.115.8] in the directory litprog.

5 Internet Nodes of Interest to Literate Programmers

The principal nodes of interest to literate programmers are the Literate Programming Archive (LPA hereafter) and the CTAN (Comprehensive T_EX Archive Network).

The Literate Programming Archive is:

```
Node:      ftp.th-darmstadt.de [130.83.55.75]
Directory: pub/programming/literate-programming
Notes:     Fastest response during off-U.S. [yep] business hours.
```

The CTAN sites are:

ftp host	IP	CTAN root
ftp.TeX.ac.UK	134.151.79.32	pub/archive
ftp.Uni-Stuttgart.DE	129.69.1.12	soft/tex
ftp.SHSU.edu	192.92.115.10	tex-archive

Institution and Sponsors: Aston Univ./UK T_EX U.G.; Univ. Stuttgart/DANTE e.V; Sam Houston State University.

Other nodes and directories of interest include:

```
Node:      niord.shsu.edu [192.92.115.8]
Directory: various (do some snooping!)
Notes:     Has a gopher server.
```

```
Node:      ftp.desy.de [131.169.10.38]
Directory: pub/web. Various documents, samples, and the FWEB FAQ.
Notes:     Has a www server, http://info.desy.de:80/
```

6 What is Literate Programming?

Literate programming is the combination of documentation and source together in a fashion suited for reading by human beings. In fact, literate programs should be enjoyable reading, even inviting! (Sorry Bob, I couldn't resist!) In general, literate programs combine source and documentation in a single file. Literate programming tools then parse the file to produce either readable documentation or compilable source. The WEB style of literate programming was created by D.E. Knuth during the development of his T_EX typesetting software.

All the original work revolves around a particular literate programming tool called WEB. Knuth says:

The philosophy behind WEB is that an experienced system programmer, who wants to provide the best possible documentation of his or her software products, needs two things simultaneously: a language like T_EX for formatting, and a language like C for programming. Neither type of language can provide the best documentation by itself; but when both are appropriately combined, we obtain a system that is much more useful than either language separately.

The structure of a software program may be thought of as a web that is made up of many interconnected pieces. To document such a program we want to explain each individual part of the web and how it relates to its neighbours. The typographic tools provided by T_EX give us an opportunity to explain the local structure of each part by making that structure visible, and the programming tools provided by languages such as C or Fortran make it possible for us to specify the algorithms formally and unambiguously. By combining the two, we can develop a style of programming that maximizes our ability to perceive the structure of a complex piece of software, and at the same time the documented programs can be mechanically translated into a working software system that matches the documentation.

7 How to begin literate programming

A recommended book is Wayne Sewell's (1989) 'Weaving a Program: Literate Programming in WEB,' Van Nostrand Reinhold, ISBN 0-442-31946-0 (pbk).

Another is D.E. Knuth's collection of articles (1992) 'Literate Programming,' Center for the Study of Language and Information, Stanford University, ISBN 0-937073-80-6 (pbk). This is more of historical interest than Sewell's book, but is a good read.

Some talk exists in the newsgroup/ mailing list for a Usenet University course in literate programming. I'm sure discussion of this topic will be welcomed. If you are interested, please participate.

8 What Tools are Available

A significant number of tools for literate programming are available. Most have been ported from their original systems, so support multiple computer platforms. If you are the developer of such a tool, and would like to make the software freely available, please send me email and I'll

reply with a form (like those below) for you to fill in. (Or short-circuit the process and kludge a form from below.)

8.1 FWEB

Developer: John A. Krommes
 System(s) supported: Unix and DOS platforms
 Software supported: C, C++, Fortran-77, Fortran-90, Ratfor, \TeX , and \LaTeX .
 Availability: anonymous ftp from lyman.ppp.l.gov/pub/fweb
 anonymous ftp from CTAN/web/fweb
 anonymous ftp from LPA/machines/ms-dos
 Readme: In bundle with above.
 Brief description: It also has a well-developed user's manual and its own FAQ (see above). It runs on most platforms: VMS, PC, UNIX, and pretty much anything that the GNU C compiler (GCC) is supported for.
 Support: bug reports to krommes@princeton.edu

8.2 CWEB

Developer: Silvio Levy and D.E. Knuth
 System(s) supported: Unix systems (dos port available)
 Software supported: C, C++, \TeX , and \LaTeX .
 Availability: anonymous ftp from labrea.stanford.edu/pub/cweb
 anonymous ftp from CTAN/web/c_cpp/cweb
 anonymous ftp from CTAN/web/c_cpp/cwb30p8c (dos version)
 Readme: bundled with above
 Brief description:
 Support: bugs to levy@math.berkeley.edu

8.3 NOWEB

Developer: Norman Ramsey <norman@bellcore.com>
 System(s) supported: Unix and DOS platforms
 Software supported: All programming languages, (\LaTeX) \TeX formatters.
 Availability: anonymous ftp from csservices.princeton.edu/pub/noweb.shar.Z
 anonymous ftp from bellcore.com/pub/norman/noweb.shar.Z
 Readme: with bundle above
 Brief description: noweb is designed to meet the needs of literate programmers while remaining as simple as possible. Its primary advantages are simplicity, extensibility, and language-independence. noweb uses 4 control sequences to WEB's 27, and its manual is only two pages. noweb works 'out of the box' with any programming language, and its formatter-dependent part is under 50 lines. The primary sacrifice relative to WEB is the loss of the language-dependent features: prettyprinting and an index of identifiers.
 Support: by email to the author

8.4 NUWEB

Developer: Preston Briggs: preston@cs.rice.edu
 Systems supported: Unix systems; Sparcs, RS/6000s, HPs, and (!) MSDOS.
 Software supported: It's written in C. It works with any programming language (or combination of programming languages) and \LaTeX

Availability: anonymous ftp from CTAN/web/nuweb (UNIX)
 anonymous ftp from CTAN/web/nuweb-pc (MSDOS)
 anonymous ftp from LPA
 Readme: Send mail to preston@cs.rice.edu
 Brief description: A single program that takes a web file written in a combination of \LaTeX and any programming language(s) and produces a \LaTeX file that can be pretty printed and a set of files containing code for compilation/interpretation by the appropriate language processors.
 Strengths include speed, simplicity, multiple languages, nice indices and cross-references, \LaTeX . Doesn't require any special macros or macro files.
 Drawbacks: \LaTeX -dependent, no code pretty printing, harder to make indices than cweb. More good stuff: nice support for make, doesn't reformat source files, so they're easy to debug. Lots of control without too much effort. That is, it doesn't do too much! Future directions... Very little change planned, except perhaps refinements in the indexing software.
 Support: Hack it yourself or send e-mail to preston@cs.rice.edu

8.5 FunnelWeb

Developer: Ross N. Williams: ross@guest.adelaide.edu.au
 Systems supported: MSDOS, Mac, VMS, Sun. Other ports reported.
 Software supported: For printing, \TeX . Otherwise no restrictions.
 Availability: anonymous ftp from CTAN/web/funnelweb
 anonymous ftp from ftp.adelaide.edu/pub/funnelweb
 Readme: With bundle above.
 Brief description: FunnelWeb is a production-quality literate-programming tool that emphasises simplicity and reliability. Everything about FunnelWeb, from the simplicity of its language to the comprehensive tutorial in the user's manual, has been designed to make this as simple, as practical, and as usable a tool as possible.
 Features:
 + Provides a simple macro preprocessor facility.
 + Can produce typeset documentation.
 + Runs on Sun, VMS VAX, Macintosh, PC, and others.
 + Portable C source code distributed under GNU licence.
 + Comprehensive user's manual including tutorial.
 + Programming-language independent.
 + Can generate multiple output files.
 + Allows complete control over the output text.
 + Regression test suite with over 200 tests.
 + Fully worked example (in /pub/funnelweb/examples).
 - Requires \TeX to produce typeset documentation.
 - Typesets program code using TT font only.
 Support: No formal support available. Mailing list maintained with about 50 subscribers. Informal assistance available from mailing list.

8.6 SchemeWEB

Developer: John D. Ramsdell
 System(s) supported: Unix and DOS platforms
 Software supported: Programming language: Any dialect of Lisp.
 Text formatter: \LaTeX .
 Availability: The Unix version is in the Scheme Repository and it is available via anonymous ftp from nexus.yorku.ca in [/pub/scheme/utl](ftp://pub/scheme/utl). The DOS version is part of the PCS/Geneva Scheme system which is available via anonymous ftp from cui.unige.ch in [/pub/pcs](ftp://pub/pcs).
 Readme: In bundle with above.
 Brief description: SchemeWEB is a Unix or DOS filter that translates SchemeWEB source into \LaTeX source or Lisp source. Lisp code is formatted in typewriter type font and the comments are passed to \LaTeX unchanged.
 Support: Send bug reports to ramsdell@mitre.org.

8.7 C2LaTeX

Developer: John D. Ramsdell
 System(s) supported: Unix
 Software supported: Programming language: C.
 Text formatter: \LaTeX but it's easy to change the formatter.
 Availability: Anonymous ftp from omnigate.clarkson.edu:[pub/tex/tex-programs/c2latex](ftp://pub/tex/tex-programs/c2latex).
 Readme: Absent. Documentation is in the C source for [c2latex](ftp://pub/tex/tex-programs/c2latex).
 Brief description: C2latex provides simple support for literate programming in C. Given a C source file in which the comments have been written in \LaTeX , [c2latex](ftp://pub/tex/tex-programs/c2latex) converts the C source file into a \LaTeX source file. It can be used to produce typeset listings of C programs and/or documentation associated with the program. C2latex produces \LaTeX source by implementing a small number of rules. A C comment that starts at the beginning of a line is copied unmodified into the \LaTeX source file. Otherwise, non-blank lines are surrounded by a pair of formatting commands (`\begin{flushleft}` and `\end{flushleft}`), and the lines are separated by `*` . Each non-blank line is formatted using \LaTeX 's `\verb` command, except comments within the line are formatted in an `\mbox`.
 Support: Send bug reports to ramsdell@mitre.org.

9 Other resources

An untapped resource (by me anyway ;-)) is the World Wide Web. Marcus Speh has expended considerable effort in this regard. If you're connected to WWW, then access:

```
http://info.desy.de/pub/www/projects/
                               LitProg.html
```

If you aren't connected to WWW, telnet to info.cern.ch and explore. You can reach Marcus' literate programming pages by typing:

```
go http://info.desy.de/pub/www/projects/
                               LitProg.html
```

Help for people who have only Email and neither WWW nor telnet, can be obtained by Email from LISTSERV@INFO.CERN.CH by sending a message containing the command HELP. Instructions will be returned by email. For literate programming documents, you can try anonymous ftp to [ftp.desy.de](ftp://ftp.desy.de) [131.169.30.33] and attach to directory:

```
/pub/www/projects/Announce
```

Another resource of interest to literate programmers is the info-tex mailing list. If you're using \LaTeX as your typesetting system and have access to internet, then you should investigate this mailing list. Mail list service is available through the SHSU list-server. To subscribe, send a message to LISTSERV@SHSU.EDU (LISTSERV@SHSU.BITNET), and include in the message one line of text:

```
SUBSCRIBE INFO-TEX 'your name in quotes'
```

The list is unmoderated; messages sent to info-tex@shsu.edu are automatically distributed to all subscribers and cross-posted to comp.text.tex.

Archives of the mailing list and newsgroup are maintained on niord.shsu.edu [192.92.115.8] in the directory info-tex.

10 Code Examples

Examples of web programs are included with the FWEB distribution. Also, demonstration programs are included with the CWEB distribution.

Bart Childs converted the K&R calculator program into a literate program. It can be retrieved from niord.shsu.edu [192.92.115.8] from directory [kr-cweb-sample](ftp://niord.shsu.edu/kr-cweb-sample) as [kr-cwsamp.zip](ftp://niord.shsu.edu/kr-cwsamp.zip).

Ross Williams has released a funnelweb example. You can retrieve this file from node [ftp.adelaide.edu.au](ftp://ftp.adelaide.edu.au) [129.127.40.3] as [/pub/funnelweb/examples/except](ftp://ftp.adelaide.edu.au/pub/funnelweb/examples/except). * This file should be on CTAN as well.

11 Bibliographies

Nelson Beebe has collected an extensive bibliography treating literate programming. His work is available for anonymous ftp from [ftp.math.utah.edu](ftp://ftp.math.utah.edu) [128.110.198.2] in directory [/pub/tex/bib](ftp://ftp.math.utah.edu/pub/tex/bib) as files:

```
litprog.bib
litprog.ltx.
```

Although I have not verified this, LPA is an alternate source for these files. Note that they are updated frequently (Nelson says several times each week), so be sure to get a fresh copy before extensive use. Joachim Schrod indicates that these files may be updated daily and can be retrieved via anonymous ftp at LPA/documentation.

12 How to anonymously ftp

Pretty much everything mentioned here is available by anonymous FTP. FAQ lists cross-posted to news.answers and rec.answers can be gotten from rtfm.mit.edu (18.70.0.224), under /pub/usenet/news.answers or under /pub/usenet/more.specific.group.name

‘anonymous FTP’ is just a way for files to be stored where anyone can retrieve them over the Net. For example, to retrieve the latest version of the literate programming FAQ, do the following:

```
> ftp rtfm.mit.edu
/* connect to the site; message fol-
lows */
> anonymous
/* type this when it asks for your name */
> <your email address>
/* type your address as the password */
> cd /pub/usenet
/* go to the directory you want to be */
> cd comp.programming.literate
/* one level down (no slash). */
> dir
/* look at what's there */
> get litprog.faq
/* get the file; case-sensitive */
> quit
/* stop this mysterious thing */
```

If your FTP program complains that it doesn't know where the site you want to use is, type the numerical address instead of the sitename:

```
> ftp 18.70.0.224
/* connect with numerical address */
```

If you don't have ftp access, send e-mail to mail-server@rtfm.mit.edu with the single word ‘help’ in the body of the message.

Getting binary files (executables, or any compressed files) is only slightly more difficult. You need to set binary mode inside FTP before you transfer the file.

```
> binary /* set binary transfer mode */
> ascii /* set back to text transfer mode */
```

FAQs and spoiler lists are generally ascii files; everything else is generally binary files.

Some common extensions on binary files in archive sites are:

.Z	Compressed; extract with uncompress
.tar.Z	Compressed ‘tape archive’; uncompress then untar or tar -xvf
.gz or .z	Gnu gzip; use gunzip (available from prep.gnu.ai.mit.edu)
.sit	(Mac) Stuffit archive
.zip	(PC) Extract with Zip or Unzip
.zoo	(PC) Yet another archive/compress program
.lhe	(Amiga) ?
.exe	(PC) Self-extracting archives - just execute them.
.uue or .UUE	Transfer as text file; use uuencode to convert to binary
.hqx	(Mac) BinHex format; transfer in text mode

Generic help can be found in the FAQs of comp.binaries.<your_system_type> for how to transfer, extract, and virus-check binary files. (At rtfm.mit.edu)

If you can't FTP from your site, use one of the following ftp-by-mail servers:

```
ftpmail@decwrl.dec.com
ftpmail@src.doc.ic.ac.uk
ftpmail@cs.uow.edu.au
ftpmail@grasp.insa-lyon.fr
```

For complete instructions, send a message reading ‘help’ to the server.

If you don't know exactly what you're looking for, or exactly where it is, there are programs and servers that can help you. For more info, send e-mail to mail-server@rtfm.mit.edu with the body of the message reading send usenet/news.answers/finding-sources

Thanks to Aliza R. Panitz (the ‘buglady’) for this text. I copied it verbatim from her post on faq-maintainers with only minor modifications.

Acknowledgements

This document would not have happened without the help of many people. Among them are Marcus Speh, George Greenwade, Rob Beezer, Joachim Schrod, Piet van Oostrum, and Ross N. Williams. A special thanks to Aliza R. Panitz for the text describing how to execute an anonymous ftp for files of interest.

Any omissions from these acknowledgements should be considered an act of stupidity on my part. Of course, the authors of literate programming tools mentioned above all play a vital role in the vitality of literate programming. Furthermore, participants in the comp.programming.literate newsgroup (and associated mailing list) all contributed in various fashions. Thank all of you.

Notes

This document will be evolving over the next few months. I'm planning on adding entries for additional literate programming tools and will expand the sections on examples as more examples become available. Tools I will include are web (the original pascal version), funnelweb, spiderweb, and cnoweb for starters. Others will be added as I find and document them. Omission of a particular tool should not be considered a snub in any sense—simply an error or oversight on my part.

History

First release: Fri 10/15/1993 – Yep, I finally did it!

Bach_oT_EX '94

April 30th–May 3th, Bach_oT_EX, Poland

Kees van der Laan

Guitars in the night

1 Introduction

This no-nonsense and low-budget meeting of the Polish T_EX User Group GUST was held at a lake in the woods. These grounds are proprietary of the Copernicus University of Toruń, and a very pleasant surrounding for a conference. The conference attracted some 70 odd participants.

On Saturday night we had our campfire with a nice barbecue and the audience was carried away into the music by guitar players. A very cosy and natural atmosphere. Relaxed, at ease, and I enjoyed it very much! The prelude and postlude of this evening were similar: each evening there was a gathering somewhere in a cabin with singing and people talking and enjoying each others company.

GUST is doing well. Two years ago I paid Hanna and Włodek a visit and there was not yet such a kind of a meeting, neither a GUST bulletin nor a list server or a file server. Congratulations GUST, go on, and we from NTG bear you a very warm heart.

2 Tutorials

The meeting started with what I consider the most important issue with T_EX: education. The Saturday and Sunday were completely devoted to tutorials¹

- T_EX surveyed (Marek Ryćko)
- Polish typographical tradition (Andrzej Tomaszewski)
- L^AT_EX₂_ε (Thomasz Przechlewski)
- L^AMeX₂e (Włodek Bzyl)
- Index and bibliography (Włodek Bzyl)
- T_EX + PostScript (Piotr Pianowski and Stanisław Wawrykiewicz)
- DVIPS, how to use it (Stanisław Wawrykiewicz)
- Fonts with T_EX and PostScript (Bogusław Jackowski)
- METAFONT—practical and non-practical examples (Bogusław Jackowski)

Each tutorial was written up in a A5 sized booklet of typographical good quality. I can only pity that they were written in Polish. Especially, the last survey of METAFONT is the best I have ever laid eyes on, and definitely

deserves translation.² The tutorial of Piotr treated similar functionalities as PStricks, but more complete and above all simpler. He definitely will be asked³ to rehearse on his presentation at EuroT_EX '94, and then we will automatically have the work in English. I have already gotten some of his macros, but he said that some documentation was in Polish, alas. But wait and see, it will be there in due time. It drew my attention, no escape ;-).

3 Conference

The Polish contributions were about their archive—yes they have got one!—L^AT_EX and Polish language, and $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX. The Polish L^AT_EX lecture sparked the awareness that the Polish people should cooperate on this and submit the results to the Babel branch of the L^AT_EX₃ project. During the meeting the people involved gathered and I promised to inform Johannes Braams about this.

Undoubtedly, because of his charism when talking Phil Taylor was invited to lecture about book design for T_EXies. 2 hours full of theory with after dinner another 3 hours with practicalities. One of his pleas is that the formatting of a book should be done with two facing pages as a unit to be shipped out. I'm very eager to reread all this, when it comes out in the GUST proceedings issue, or so.

The Tuesday morning was completely offered to me. Of course I was very honoured to be invited by this group with such a number of outstanding T_EXies. Off-the-self I had taken with me Manmac BLUes, and the tune was well received: if only there had been a User's Guide. . .

After coffee I could also tell about BLUE, how he copes with bibliographies, transparencies and verbatims.

All was warmly received and I was invited to work on the Manmac User's Guide, next to submitting my articles and macros to their archive.⁴ The only problem is that writing a real and attractive User's Guide is quite something, and requires skills complementary to mine. Moreover, I had made plans already to work on METAFONT in three dimensions, to emulate Naum Gabo's works.

At the end of the conference I was invited to select from the filled-in questionnaires a 'winner.' After some theat-

¹ Some of the titles of the tutorials have been 'translated' by me. I apologize for the incorrections, but the flavour is definitely correct because I was there. And Polish friends, don't blame me for not using your fancy fonts. I'm sorry for the moment. It will be on the 4allT_EX CD-ROM, so another time.

² I hope so much that Julita will actively join GUST and starts cooperating with Jackowski to translate this nice material.

³ In fact he already agreed, but he was a bit modest about his qualities as a lecturer, but we can handle that can't we?

⁴ Thank you!

rical acts hilarity was all around when I could receive from Hanna the bar of chocolate. Once again thank you all, I enjoyed it very much, not in the least because I made some real new friends.

4 EuroTeX '94

A side-effect of this meeting was that preparations for EuroTeX '94 progressed very much. Włodek and GUST can't do it on their own. Hanna, Phil and me are on the program committee. All LUG presidents will be invited to join the committee of trustees, to help as much as they can if not for stimulating their members to join that conference and meet the Poles (and others) themselves. The conference location is well-chosen: a former party-bosses residence at sea on one of the grounds in the delta of the Wisła. The conference theme is

TeX: principles, practice and Poland

Hope to see you there and let us face the music.

5 Before and after

I headed for Poland a week earlier and enjoyed the hospitality of the Bzyl family. Włodek and I did some work, talked not only about TeX but also about Knuth's works in general, next to discussing the usefulness of some other (UNIX-like) tools. Last time I was impressed by the beauty of Sopot. Now I enjoyed Gdynia, especially the pier and the shopping area.

After the meeting I received hospitality of Krzys' family,⁵ and I must say that I'll never forget the revisiting of the Wilanowa. A real romantic place. Also my goodbye evening with a nice dinner with Krzys, Hanna and Julita, followed up by enjoying the opera, will pop up now and then in my sweet dreams for sure.

P.S.

Their list address: GUST-L@vm.cc.uni.torun.pl.

The fileserver and ftp address at: halina.univ.gda.pl.

⁵Krzys gave me a copy of the 'The Euler project at Stanford, by D.R. Siegel,' and Julita⁶ shared her Chinese fonts with me. Definitely, something for the next CD-ROM.

NTUG '94

May 16th, Oslo, Norway

Kees van der Laan

METApost Knuth says

1 Introduction

The week before Knuth passed by on his way to St. Petersburg where he was invited for another honorary degree¹. At Oslo he lectured about his graphbase work. My host, Dag Langmyhr, asked him what he used himself nowadays. Knuth answered: $\text{T}_{\text{E}}\text{X}$, CWEB , METApost , and another one which I forgot. Dahl—yes, the one from $\text{SIMUALA}'67$ —a personal friend of Knuth still works at the University of Oslo, the meeting place of NTUG '94. I left Garnwerd on Saturday and enjoyed a very natural and hospitable stay at Dag's place.

2 The meeting

Sober it was. Not that many attendees, a dozen or so. In circulating the meeting place over Denmark, Sweden, Norway and Finland, those interested to attend have to spend quite some money. Therefore it is understandable that not that many did show up, alas. Even from within Norway some did come by plane.

Lars Thorensen started with reporting about his experience in 'Creating hyphenation patterns for Norwegian.'² He mentioned a succes rate of $\approx 90\%$, with Liang's pattern technique.

Next, Dag himself lectured about 'How to create your own documentstyle?' A very clear and concise survey of what you are likely to modify when customizing $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$'s report style. He considers adapting his lecture towards $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}2\epsilon$, and to submit this for $\text{EuroT}_{\text{E}}\text{X}'94$. Definitely worthwhile. It occurred to me that the NTUG is $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ biased.

Rolf Lindgren, a psychology student, had worked on 'Explorations in typesetting Russian-Icelandic and Russian-Norwegian dictionaries.' His results suffered from ample 'buses-and-weirdness'-effects and Roswitha suggested to use a wider inter-column space next to the use of `\raggedright`. Sensible.

After lunch the floor was mine, and I presented TUGboat BLUEs for the very first time, stressing the insights aspects by questioning various items and contrasting the solutions with alternatives. The discussion inspired me towards what to submit to $\text{EuroT}_{\text{E}}\text{X}'94$. It will be BLUE's format, not only a user's guide to manmac, but also to my `tug.ppt` proposal, and to macros in general provided by the $\text{T}_{\text{E}}\text{X}$

community at large. Flexible with respect to the inner markup of math, tables and the like. The basic idea is to distinguish two markup levels

- the outer level which accounts for the title, (sub)sections, appendices etc., and
- the inner level which accounts for document elements like math and tables to be positioned within the context.

The important aspect is that the inner level allows for generic markup, meaning that it should be independent of the used outer level, be it plain, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, or you name it. The audience for this work is the $\approx 90\%$ unorganized $\text{T}_{\text{E}}\text{X}$ users, who don't care so much about all those varieties of $\text{T}_{\text{E}}\text{X}$ markup, and just use plain $\text{T}_{\text{E}}\text{X}$ with some add-ons, to get their work out. BLUE's format aims at being such an add-on, and flexible with respect to incorporating any suitable macro written in plain.

Then there was the annual meeting where the status of NTUG was discussed.

After the meeting we had a very cosy informal dinner in one of Oslo's ample harbour restaurants.

3 The day after

May in Norway is special, not in the least because of the 17th, when the Norwegians celebrate the emerge of their constitution. A very peaceful happening where children's parades dominate the scene. The Norwegians love their children, and are aware that they have the future. Very natural and peaceful. This social program was really something. I enjoyed it very much.

Jan-Michael, Dag and I discussed among others ways to guarantee that lines in facing pages or in neighbouring columns line up. Dag suggested the very elegant solution to allow for some kind of discrete glue, to allow glue to be inserted such that the position to start a new line equals an integral number of line distances since the beginning of the page, independent of what has been inserted. The only problem is that $\text{T}_{\text{E}}\text{X}$ is frozen.

Another aspect which struck me was that for communication KTH considers radio links instead of telephone-modem connections.³

Nordic $\text{T}_{\text{E}}\text{X}$ Users thank you for having invited me, to provide the opportunity to discuss things. Dag and Anne-Bret thank you for your kind hospitality. It was a great pleasure!

¹Irina Makhovaya reported that it was a thrill to have Knuth over in Russia, and to discuss things with him.

²Those not familiar with the history of the Nordic countries, please note that in Norway they have two languages, and that the common language for the NTUG is English.

³KTH stands for the Royal Institute of Technology for Sweden, and is the place which employs Roswitha and Jan-Michael.

Announcement TUG'94

Santa Barbara, California
31 July – 4 August

By now most of you will have received the pink flyer with preliminary information on the conference: the presentations are outlined on the next page, the courses are listed on p. 26. Some points to highlight include:

1. The conference and accommodation fees cover the period from **Saturday evening** (30 July) to **Thursday afternoon** (4 August).
2. **Registration** begins on Saturday; the **Reception** will be held that evening.
3. The **Banquet** is *included* in the accommodations fee (for those staying off-campus there will be a \$35 charge).
4. **News Flash!** There will be a one-day registration fee of 100.00 (can only be used once).

Bursary Fund

This will be the second year for TUG's Bursary Fund. Although some funds were left over from last year, we are taking this opportunity to let the membership know that the current Bursary Committee will be inviting all TUG members to consider contributing to the fund. TUG'94 registration forms will provide a space for donations; if you are unable to attend, but would still like to contribute to the fund, you may do so by contacting the TUG office directly for more information. The Bursary Committee comprises Bernard Gaille and Norman Naugle, with Christina Thiele as liaison to the board.

The bursary fund is intended to provide financial assistance to cover travel and/or expenses to attend the upcoming TUG meeting in Santa Barbara. The fund is not focused on any specific geographic region; rather, the fund is for those who demonstrate reasonable need, and/or have never attended a TUG meeting before.

If you wish to apply for a bursary, kindly send a letter to Bernard Gaille, chair of the Bursary Committee, providing details of your requirements, including precise costs where possible. Bernard Gaille, 4, avenue Cadoux-Girault, F-92270 Bois-Coulombes, France; e-mail: gaille@idris.fr.

Preliminary outline program

Sunday July 30th: Publishing, languages, literature and fonts

- ▷ Glenn Reid: *Reflections*
- ▷ Frank Mittelbach: *Real Life Book Production*

- ▷ Yannis Haralambous: *Typesetting the Holy Bible in Hebrew, with T_EX*
- ▷ Basil Malyshev: *Automatic Conversion of METAFONT Fonts to Type1 PostScript*
- ▷ Alan Jeffrey: *PostScript Font Support in L^AT_EX 2_ε*
- ▷ Y. Haralambous: *An Indic T_EX Preprocessor — Sinhalese T_EX*
- ▷ Michel Cohen: *Zebrackets: A metaMETAFONT*

Monday August 1st: Colour and L^AT_EX

- ▷ Leslie Lamport: *Looking Back At, And Forward From, L^AT_EX*
- ▷ Tom Rokicki: *Advanced 'Special' Support in a dvi Driver*
- ▷ Timothy van Zandt and Denis Girou: *An Introduction to PStricks*
- ▷ Sebastian Rahtz and Michel Goossens: *Simple Colour Design in T_EX*
- ▷ Michael Sofka: *Color Book Production Using T_EX*
- ▷ James Hafner: *The (Pre)History of Color in Rokicki's dvips*
- ▷ Friedhelm Sowa: *Printing Colour Pictures*
- ▷ Angus Duggan: *Colour Separation and PostScript*
- ▷ Jon Stenerson: *A L^AT_EX Style File Generator*
- ▷ Johannes Braams: *Document Classes and Packages in L^AT_EX 2_ε*

Tuesday August 2nd: T_EX Tools

- ▷ Oren Patashnik: *BibT_EX 1.0*
- ▷ Minato Kawaguti and Norio Kitajima: *Concurrent Use of Interactive T_EX Previewer with an Emacs-type Editor*
- ▷ Y. Haralambous: *Humanist*
- ▷ Pierre Mackay: *A Typesetter's Toolkit*
- ▷ Jean-luc Doumont: *Pascal Pretty-printing: An Example of 'Preprocessing within T_EX'*
- ▷ Michael P. Barnett and Kevin R. Perry: *Symbolic Computation for Electronic Publishing*
- ▷ Norm Walsh: *A World Wide Web Interface to CTAN*

Wednesday August 3rd: Futures

- ▷ Chris Rowley and F. Mittelbach: *The Floating World*
- ▷ Joachim Schrod: *Towards Interactivity for T_EX*
- ▷ Arthur Ogawa: *Object-oriented Programming, Descriptive Markup, and T_EX*
- ▷ William Erik Baxter: *An Object-oriented Programming System in T_EX*
- ▷ John Plaice: *Progress in the Omega Project*
- ▷ Philip Taylor: *e-T_EX & NTS: A Progress Report*
- ▷ Jonathan Fine: *Documents, Compuscripts, Programs and Macros*
- ▷ George Greenwade: *T_EX as a Commodity*

Thursday August 4th: Publishing and design

- ▷ Maurice Laugier and Y. Haralambous: *T_EX Innovations by the Louis-Jean Printing House*
- ▷ Michael Downes: *Design by Template in a Production Macro Package*
- ▷ Gabriel Valiente Feruglio: *Macro Packages for Typesetting Commutative Diagrams*

- ▷ Alan Hoenig: *Less is More: Restricting T_EX's Scope Enables Complex Page Layouts*
- ▷ Don Hosek: *Sophisticated Page Layout with T_EX*
- ▷ Henry Baragar and Gail E. Harris: *An Example of a Special Purpose Input Language to L^AT_EX*
- ▷ Marko Grobelnik, Dunja Mladenić, Darko Zupanič and Borut Žnidar: *Integrated System for Encyclopaedia Typesetting Based on T_EX*
- ▷ Y. Haralambous and J. Plaice: *First Applications of Ω: Adobe Poetica, Arabic, Greek, Khmer*

Posters, workshops and discussion sessions

1. The TUG WWW Server (Peter Flynn)
2. Accessing CTAN (N. Walsh)
3. Practical Indexing (Nelson Beebe)
4. T_EX and Linguistics (Christina Thiele)
5. T_EX and Humanities Journals (C. Thiele)
6. Database Publishing (M. Grobelnik)
7. Floats (David Salomon)
8. Adobe Acrobat and Related Technologies (S. Rahtz)
9. Standards for Colour Drivers (T. Rokicki)

INNOVATIONS!**TUG 15th Anniversary Meeting**

Where?	Santa Barbara, California (west northwest of Los Angeles)
When?	July 31 to August 4 (with courses before/after these dates)
Who?	Everyone interested in T _E X, L ^A T _E X, and METAFONT!

Announcement EuroT_EX '94

Sobieszewo, Poland ,
26 – 30 September 1994

Is North America too far or too expensive or too alien a culture? Then come to EuroT_EX '94, *the* T_EX conference of the year. EuroT_EX'94 will take place at Sobieszewo on an idyllic island off the coast of Gdansk in Poland. The conference will run from Monday September 26th to Friday September 30th, and the *maximum* cost (based on two persons sharing) will not exceed \$260 or £ 175 or DM 450. Those arriving early on Monday will be able to take part in a guided tour of the old town of Gdansk, whilst Tuesday to Thursday will be jam-packed with talks and tutorials on T_EX and related topics.

All delegates will be accommodated in a single building, and for the whole week will be cut off from civilisation: no distractions, no need to leave the island: everything will be provided. For those unable to sustain the pace, quiet meditative walks along the shore searching for amber will provide the ideal opportunity for therapeutic meditation.

Papers are solicited *now*, and early registration for the conference is advised: with its Central European location and idyllic setting, the conference is expected to attract many delegates. If you are even *thinking* of coming, then you are urged to contact us to be added to the mailing list.

Please note the following deadlines:

Abstracts for papers	:	no later than 1 May 1994;
Final papers	:	15th August 1994;
Provisional registration	:	1 June 1994 (no charge for cancellation)
Confirmed registration	:	1 September 1994 (cancellation charged at 50%)
Late registration	:	15th September 1994 (no cancellation possible)

Bursaries

Similar to the TUG meeting at Aston last year, it is hoped to be able to offer financial assistance to delegates who would otherwise be unable to attend; of course, we cannot be sure at this stage that sponsors will be as generous as they were last year, but intending delegates who will need assistance in order to be able to attend should indicate in the space provided the *minimum* bursary which would allow them to be able to attend, and should give clear reasons why they are applying. All applications will be treated in the strictest confidence. Delegates who are in no need of a bursary and who are able to assist others less fortunate are urged to pledge a donation.

Tutorials and Courses

It is intended to offer both tutorials (which will take place during the week of the conference proper), and courses (which will take place during the week following the conference); proposed topics include book design and L^AT_EX₂_ε, but no firm decisions have yet been taken on topics, durations or costs. Further details concerning this area will be circulated as soon as they are known.

Correspondence

Sent queries to EuroT_EX '94 Conference Organisers, c/o Włodek Bzyl, Department of Mathematics, University of Gdansk, Poland; e-mail: Wlodek Bzyl (Mathematics) eurotex@halina.univ.gda.pl

Announcement: 4all \TeX CD-ROM

Erik Frambach & Wietse Dol

4 \TeX : a \TeX Workbench for MS-DOS PC's

\TeX and all its companions offer an enormous amount of possibilities. This is both an advantage and a disadvantage. The advantage is that almost anything is possible. The disadvantage is that you need detailed knowledge of all related programs to fully exploit the possibilities.

4 \TeX is an attempt to integrate all major \TeX related programs in a shell that shields you from the tedious and frustrating job of setting environment variables and program parameters.

In 4 \TeX all functions are available through simple menus, so profound knowledge of all underlying programs is no longer required.

4all \TeX : the NTG distribution set

At the end of 1993 the NTG (the Dutch language oriented \TeX users group) released the package 4all \TeX . It consists of a set of 31 diskettes, on which the most important \TeX and \LaTeX related packages are assembled in well structured modules. The set is especially interesting to MS-DOS users because it contains the complete 4 \TeX workbench. Others will also find it valuable because of the 'public domain' system independent parts of it. This includes of course \TeX and \LaTeX files, but also PostScript fonts.

4all \TeX : the NTG CD-ROM

To simplify the distribution of 4all \TeX , a CD-ROM will be released in June! This CD-ROM should enable you to set up a 4 \TeX system with minimal effort.

The CD-ROM will *at least* contain:

- a fully installed 4 \TeX 3.20 workbench, ready for use;
- the complete 4all \TeX distribution set;
- the MAPS (NTG's 'Minutes & Appendici', articles on \TeX related topics, mostly in English) of all years in PostScript format;
- all TEX-NL, UK-TEX, TEXHAX and TEXMAG mailing list discussions of recent years;
- extra documentation, such as:
 - 'The \TeX book' and 'The METAFONT book' in \TeX code,
 - \TeX , \LaTeX and METAFONT introductions,
 - very extensive bibliographies on miscellaneous \TeX related topics in BiB \TeX format,
 - indexes to \TeX macros and to files on the Comprehensive \TeX Archive Network;

- extra programs, such as:
 - GNUplot,
 - GhostScript,
 - dvi utilities,
 - PostScript utilities,
 - Polish and Cyrillic \TeX packages,
 - WEB: literate programming.

The CD-ROM will include the updated documentation on 4 \TeX , an A5 sized booklet of approx. 120 pages, that explains how 4 \TeX works in depth. The price of the CD plus booklet is set at Dfl. 60,- or US \$ 35.

The CD will be put out in a limited edition, and no reprint is scheduled as yet.

Therefore it is wise to make reservations.

For users in The Netherlands and Belgium this is done by sending e-mail to the NTG secretary (P.O. Box 394, NL-1740 AJ Schagen, The Netherlands) or the NTG board (cgl@risc1.rug.nl), specifying how many CD-ROMs you want. Those who pay *before April 15* (postgiro 1306238, att. penningmeester NTG, Zoetermeer, The Netherlands, specifying 'CD-ROM' and the desired number) have first rights.

Orders from outside The Netherlands are in principle only accepted from the local user groups. These groups are requested to contact the NTG board (cgl@risc1.rug.nl) also before April 15. For local users groups there will be a special high volume discount.

— * —

Directory listing 4all \TeX CD-ROM

4ALLTEX	3-08-94	15:54	NTG's 4allTeX distribution set
4TEXLIST	4-18-94	15:02	4TeX discussion list, 1993-1994
AMS	3-28-94	10:49	TeX/LaTeX stuff of Amer. Math. Society
ARABIC	4-29-94	11:19	Arabic TeX
BBS	3-21-94	13:27	bulletin board info and programs
BIBLIO	3-17-94	10:46	bibliographies on TeX etc. (BibTeX)
DVIUTILS	3-14-94	9:31	programs for DVI manipulations
EDITORS	4-29-94	8:05	ASCII editors for TeX
EMTEX	3-08-94	15:54	complete 4TeX Workbench, ready to run
GNUPLOT	3-09-94	16:03	plotting program with TeX output
GREEK	4-29-94	11:18	Greek TeX
INFO	4-18-94	17:00	TeX etc. information and FAQs
LITPROG	3-14-94	9:30	Literate Programming stuff
MAPS	3-08-94	15:54	NTG's MAPS
NCDTREE	5-02-94	15:40	Tree index for Norton's NCD
NTG	3-22-94	13:52	NTG public relations set
PC_GUT	4-26-94	15:25	Gutenberg French TeX installation
POLISH	3-29-94	16:16	MEX & LaMeX, Polish TeX
PSUTILS	3-14-94	9:31	PostScript utility programs
RUSSIAN	3-29-94	16:20	Russian TeX
SPECIAL	4-29-94	11:20	special applications of TeX
TEXCOURS	3-14-94	9:32	courses on TeX, LaTeX, Metafont

```

TEXHAX 4-18-94 13:08 Texhax discussion list, 1986-1994
TEXMAG 5-02-94 9:13 TEXMAG discussions
TEXSHELL 3-14-94 9:32 another popular TeX shell
TEX_NL 3-08-94 15:54 TEX-NL discussion list, 1989-1994
TOOLS 4-19-94 16:41 tools for managing files & archives
UKTEX 4-15-94 15:55 UKTEX discussion list, 1987-1994
VDLAAN 3-15-94 15:33 Van der Laan's articles on TeX

```

Directory of \4alltex*.*

```

-----
DISK01 3-08-94 16:27
DISK02 3-08-94 16:27
DISK03 3-08-94 16:27
DISK04 3-08-94 16:27
DISK05 3-08-94 16:27
DISK06 3-08-94 16:27
DISK07 3-08-94 16:27
DISK08A 3-08-94 16:27
DISK08B 3-08-94 16:28
DISK09 3-08-94 16:27
DISK10 3-08-94 16:27
DISK11 3-08-94 16:27
DISK12 3-08-94 16:27
DISK13 3-08-94 16:27
DISK14 3-08-94 16:27
DISK15 3-08-94 16:27
DISK16 3-08-94 16:27
DISK17 3-08-94 16:27
DISK18 3-08-94 16:27
DISK19 3-08-94 16:28
DISK20 3-08-94 16:28
DISK21 3-08-94 16:28
DISK22 3-08-94 16:28
DISK23 3-08-94 16:28
DISK24 3-08-94 16:28
DISK25 3-08-94 16:28
DISK26 3-08-94 16:28
DISK27 3-08-94 16:28
DISK28 3-08-94 16:28
DISK29 3-08-94 16:28

```

Directory of \ams*.*

```

-----
AMSFONTS 3-28-94 10:51 AMS fonts
AMSLATEX 3-28-94 11:06 AMS LaTeX
AMSTEX21 3-28-94 10:54 AMS TeX

```

Directory of \ams\amsmfont*.*

```

-----
CMFONTS 3-28-94 10:51 Computer Modern fonts
CYRILLIC 3-28-94 10:51 Cyrillic fonts
DOC 3-28-94 10:52 documentation
DPI300 3-28-94 10:59
DPI329 3-28-94 10:59
DPI360 3-28-94 11:00
DPI432 3-28-94 11:01
DPI518 3-28-94 11:02
DPI622 3-28-94 11:03
DPI746 3-28-94 11:04
EUEX 3-28-94 10:51 Euler extended fonts
EULER 3-28-94 10:52 Euler fonts
SYMBOLS 3-28-94 10:51 symbol fonts
TFM 3-28-94 11:06 font metrics

```

Directory of \ams\amslatex*.*

```

-----
DOC 3-28-94 11:07 documentation
FONTSEL 3-28-94 11:07 font selection scheme
INPUTS 3-28-94 11:07 LaTeX input files
TFM 3-28-94 11:08 fonts metrics

```

Directory of \ams\amstex21*.*

```

-----
AMSTEX 3-28-94 10:55 TeX input files
DOC 3-28-94 10:55 documentation
TFM 3-28-94 10:55 font metrics

```

Directory of \arabic*.*

```

-----
FONTS 4-29-94 11:19
MFINPUT 4-29-94 11:19
REPORT 4-29-94 11:19
TEXINPUT 4-29-94 11:20

```

Directory of \arabic\fonts*.*

```

-----
300DPI 5-02-94 17:46
329DPI 5-02-94 17:46
360DPI 5-02-94 17:46
432DPI 5-02-94 17:46
518DPI 5-02-94 17:46
622DPI 5-02-94 17:46
746DPI 5-02-94 17:46
TFM 5-02-94 17:48

```

Directory of \editors*.*

```

-----
EMACS 4-29-94 8:07 Emacs (freeware)
MULTEDIT 4-29-94 8:09 MultiEdit (commercial) demo version
QEDIT30 5-02-94 18:43 Qedit 3.0 (shareware)
QEDIT215 4-29-94 8:06 Qedit 2.15 (shareware)
REDIT21 4-29-94 9:53 Redit 2.1 (freeware)
TSE 4-29-94 8:14 TheSemwareEditor (commercial) mac-
ros only

```

Directory of \editors\emacs*.*

```

-----
AUCTEX 5-02-94 9:26 LaTeX environment for Emacs

```

Directory of \editors\emacs\auctex*.*

```

-----
DOC 5-02-94 9:26 documentation
LACHECK 5-02-94 9:26 LaTeX syntax checker
STYLE 5-02-94 9:26 style files

```

Directory of \editors\redit21*.*

```

-----
MAK 4-29-94 9:54
ZUSATZ 4-29-94 9:54

```

Directory of \emtex*.*

```

-----
BTEXFMFS 3-08-94 15:58 big TeX format files
BTM 3-08-94 15:58 4TeX batch files
COMPILER 3-08-94 15:58 TeX compilers
DATA 4-14-94 17:23 emTeX dvi driver input
DOC 3-08-94 15:59 TeX & programs documentation
FORMATS 3-08-94 15:59 files needed for TeX format generation
INPUT2E 4-29-94 13:14
LOG 3-08-94 15:59 4TeX usage logging
METAFONT 3-08-94 15:59 Metafont
OS2 4-14-94 14:19 IBM OS/2 drivers
PRNDEST 3-08-94 15:59 printer types and print destinations
PS 3-08-94 15:59 PostScript driver
SPELL 3-08-94 15:59 spell checker & dictionaries
TEXFMTS 3-08-94 15:58 TeX format files
TEXFONTS 3-08-94 15:59 font files (.PK & .FLI)
TEXINPUT 3-08-94 15:59 TeX input files
TEXSAMPL 3-08-94 15:59 sample files
TFM 3-08-94 15:59 fonts metrics
UTILS 3-08-94 15:59 utility programs
WINDOWS 4-15-94 10:09 MS-Windows dvi driver

```

Directory of \emtex\btm*.*

```

-----
CONVERT 3-08-94 16:03
UTILS 3-08-94 16:03

```

Directory of \emtex\formats*.*

```

-----
SPRINGER 4-29-94 14:22 Installation files Springer Verlag

```

Directory of \emtex\metafont*.*

```

-----
BMFBASES 3-08-94 16:06 big Metafont bases
MFBASES 3-08-94 16:06 Metafont bases
MFINPUT 3-08-94 16:06 Metafont input files
MFJOB 3-08-94 16:06 Metafont job files

```

Directory of \emtex\os2*.*

```

-----
DLL 4-14-94 14:23 Dynamic Link Libraries

```

Directory of \emtex\ps*.*

```

-----
FONTS 3-08-94 16:09 PostScript fonts
REENCODE 3-08-94 16:09 alternative font encodings

```

Directory of \emtex\spell*.*

```

-----
ITALIAN 4-21-94 16:59 Italian version of amSpell programs

```

Directory of \emtex\texfonts*.*

```

-----
AFM_EXTR 3-08-94 16:01
EPSONFX 4-18-94 9:29 Epson FX fonts
FAX 4-18-94 9:29 FAX fonts
HPLASER 4-18-94 9:28 HP Laserjet fonts
LJIV 4-18-94 9:28
LQHIRE 4-18-94 9:29 Epson LQ hires fonts
LQLORES 4-18-94 9:29 Epson LQ lores fonts
VF 3-08-94 16:01 virtual fonts

```

Directory of \emtex\texfonts\epsonfx*.*			311DPI	4-18-94	9:43
-----			373DPI	4-18-94	9:43
120DPI	4-29-94	11:06	448DPI	4-18-94	9:43
170DPI	4-29-94	11:06	Directory of \emtex\texinput*.*		
180DPI	4-29-94	11:06	-----		
192DPI	4-18-94	9:42	BABEL	3-08-94	16:10 Babel language support
216DPI	4-18-94	9:42	BIBINPUT	4-29-94	10:15
240DPI	4-18-94	9:40	JOURNALS	4-29-94	10:13
263DPI	4-18-94	9:40	LANGUAGE	4-29-94	10:19
288DPI	4-18-94	9:40	NFSS	3-08-94	16:10 New Font Selection Scheme files
346DPI	4-18-94	9:40	PS	3-08-94	16:10 PostScript input files
415DPI	4-18-94	9:40	Directory of \emtex\tfm*.*		
498DPI	4-18-94	9:40	-----		
597DPI	4-18-94	9:40	ARABTEX	4-29-94	11:13
Directory of \emtex\texfonts\fax*.*			CCHESS	4-29-94	11:13
-----			CHESS	4-29-94	11:13
102DPI	4-29-94	11:04	GO	4-29-94	11:13
144DPI	4-29-94	11:04	GREEK	4-29-94	11:13
153DPI	4-29-94	11:04	MUSIC	4-18-94	18:31
163DPI	4-18-94	9:48	POLISH	4-15-94	10:44
184DPI	4-18-94	9:48	PSFONTS	4-29-94	11:13
204DPI	4-18-94	9:47	RUSSIAN	4-15-94	10:44
223DPI	4-18-94	9:47	SIMPSON	4-29-94	11:13
245DPI	4-18-94	9:47	Directory of \emtex\utils*.*		
294DPI	4-18-94	9:47	-----		
353DPI	4-18-94	9:47	COMMERC	3-08-94	16:12 commercial software
423DPI	4-18-94	9:47	EMX	3-08-94	16:12 EMX dos extender
508DPI	4-18-94	9:47	GHOSTSCR	3-08-94	16:12 GhostScript, PostScript interpreter
Directory of \emtex\texfonts\hplaser*.*			RSX	3-08-94	16:12 RSX dos extender
-----			SHAREWAR	3-08-94	16:12 shareware
150DPI	4-29-94	10:54	SOURCE	3-08-94	16:12 sources of utility programs
180DPI	4-29-94	15:53	Directory of \emtex\utils\rsx*.*		
212DPI	4-29-94	10:50	-----		
225DPI	4-29-94	10:54	BIN	3-08-94	16:12 binaries
240DPI	4-18-94	9:31	DOC	3-08-94	16:12 documentations
255DPI	4-29-94	10:54	FPU_EMU	3-08-94	16:12 floating point coprocessor emulator
263DPI	4-29-94	15:53	Directory of \emtex\utils\sharewar*.*		
270DPI	4-18-94	9:31	-----		
288DPI	4-29-94	15:54	4DOS	3-08-94	16:12 4DOS, replacement for command.com
300DPI	4-18-94	9:30	GWS	3-08-94	16:12 Graphics Workshop, bitmap convertor
305DPI	4-29-94	10:54	Directory of \gnuplot*.*		
329DPI	4-18-94	9:30	-----		
330DPI	4-29-94	15:56	DEMO	3-09-94	16:04 demo programs
360DPI	4-18-94	9:30	DOC	3-09-94	16:04 documentation
432DPI	4-18-94	9:30	Directory of \greek*.*		
480DPI	4-29-94	10:54	-----		
510DPI	4-29-94	15:56	DOC	4-29-94	11:19
518DPI	4-18-94	9:30	EMTEX	4-29-94	11:19
600DPI	4-29-94	15:56	INSTALL	4-29-94	11:19
622DPI	4-18-94	9:31	LATEX	4-29-94	11:19
746DPI	4-18-94	9:31	MF	4-29-94	11:19
Directory of \emtex\texfonts\ljjv*.*			PKFILES	4-29-94	11:18
-----			TEX	4-29-94	11:19
300DPI	4-29-94	11:01	TFM	4-29-94	11:19
424DPI	4-29-94	11:01	UNIX	4-29-94	11:19
450DPI	4-29-94	11:01	UTILS	4-29-94	11:19
480DPI	4-18-94	9:35	Directory of \greek\pkfiles*.*		
540DPI	4-18-94	9:35	-----		
600DPI	4-18-94	9:34	225DPI	4-29-94	11:18
657DPI	4-18-94	9:34	240DPI	4-29-94	11:18
720DPI	4-18-94	9:34	270DPI	4-29-94	11:18
864DPI	4-18-94	9:34	300DPI	4-29-94	11:18
1037DPI	4-18-94	9:34	328DPI	4-29-94	11:18
1244DPI	4-18-94	9:35	329DPI	4-29-94	11:18
1493DPI	4-18-94	9:35	360DPI	4-29-94	11:18
Directory of \emtex\texfonts\lqhires*.*			432DPI	4-29-94	11:18
-----			518DPI	4-29-94	11:18
180DPI	4-29-94	11:12	538DPI	4-29-94	11:19
255DPI	4-29-94	11:12	622DPI	4-29-94	11:19
270DPI	4-29-94	11:12	746DPI	4-29-94	11:19
288DPI	4-21-94	9:03	Directory of \litprog*.*		
324DPI	4-21-94	9:04	-----		
360DPI	4-18-94	9:44	CWEB	3-22-94	8:31 CWeb C-language support
394DPI	4-18-94	9:44	Directory of \litprog\cweb*.*		
432DPI	4-18-94	9:44	-----		
518DPI	4-18-94	9:44	EXAMPLES	3-22-94	8:32
622DPI	4-18-94	9:44	Directory of \maps*.*		
746DPI	4-18-94	9:44	-----		
896DPI	4-18-94	9:44	MAPS01	4-29-94	10:41
Directory of \emtex\texfonts\lqlores*.*			Directory of \maps*.*		
-----			-----		
90DPI	4-29-94	11:08	Directory of \maps*.*		
127DPI	4-29-94	11:08	-----		
135DPI	4-29-94	11:08	Directory of \maps*.*		
144DPI	4-18-94	9:43	-----		
162DPI	4-18-94	9:43	Directory of \maps*.*		
180DPI	4-18-94	9:42	-----		
197DPI	4-18-94	9:43	Directory of \maps*.*		
216DPI	4-18-94	9:43	-----		
259DPI	4-18-94	9:43	Directory of \maps*.*		
-----			-----		

MAPS02	4-29-94	10:41	HORAK	3-28-94	10:21	by Horak
MAPS03	4-29-94	10:41	KNUTH	3-28-94	10:22	The TeXbook & The Metafont book, sources
MAPS04	4-29-94	10:42	MALTYBY	4-18-94	15:17	by Maltby
MAPS05	4-29-94	10:42	PARTL	4-18-94	15:15	by Partl
MAPS06	4-29-94	10:42	SALOMON	4-18-94	18:15	by Salomon
MAPS07	4-29-94	10:43	TOBIN	4-18-94	15:21	by Tobin
MAPS08	4-29-94	10:40	VDLAAN	4-18-94	15:22	by Van der Laan
MAPS09	4-29-94	10:44	WARBRICK	4-18-94	15:12	by Warbrick
MAPS10	4-29-94	10:40	WHITNEY	4-18-94	15:32	by Whitney
MAPS11	4-29-94	10:39	WILKINS	4-18-94	15:07	by Wilkins

Directory of \ntg*.*

```
-----
4TEX      3-22-94  13:52 4TeX intro
BRIDGE    3-22-94  13:52 Bridge example
HORAK     3-22-94  13:52 Horak's Metafont examples
INTROTEX  3-22-94  13:52 introduction to TeX
MATH      3-22-94  13:52 math in TeX
MUSIC     3-22-94  13:52 MusicTeX example
NTG       3-22-94  13:52 info on NTG
SCHAAK    3-22-94  13:52 Chess example
TABLES    3-22-94  13:52 tables in TeX
WATISTEX  3-22-94  13:52 intro What Is TeX
```

Directory of \polish*.*

```
-----
EMINST    3-29-94  16:16 installation files
ISTYLES   3-29-94  16:16 Polish style files
MEXINFO   3-29-94  16:17 info on MeX
MEXLAMEX  3-29-94  16:17 LaMeX, Polish LaTeX
ORIGINAL  3-29-94  16:17 original files
PLFONTS   3-29-94  16:18 Polish fonts
TFM       4-06-94  9:02 font metrics
```

Directory of \russian*.*

```
-----
DOC        4-22-94  14:53 documentation
MAKEFONT   4-22-94  15:09 files for font making
```

Directory of \special*.*

```
-----
BRIDGE     4-29-94  11:20 Typesetting Bridge
CALENDAR   4-29-94  11:20 Typesetting calendars
CCHESSE    4-29-94  11:21 Typesetting Chinese Chess
CHEMTEX    4-29-94  11:20 Typesetting chemical structures
CHESS      4-29-94  11:20 Typesetting Chess
COOKBOOK   5-02-94  9:20 Cook book written in TeX
CROSSWRD   4-29-94  11:20 Typesetting crossword puzzles
GO         4-29-94  11:20 Typesetting Go
MUSIC      4-29-94  11:20 Typesetting music
SIMPSON    4-29-94  11:21 Typesetting The Simpsons
```

Directory of \special\music*.*

```
-----
FONTS      4-29-94  16:38 MusicTeX fonts
UTILS      4-29-94  16:38 utility programs: MIDI to MusicTeX
```

Directory of \special\music\fonts*.*

```
-----
HPLASER    4-29-94  16:38
MFINPUT    4-29-94  16:38
TFM        4-29-94  16:38
```

Directory of \special\music\fonts\hplaser*.*

```
-----
300DPI     4-29-94  16:38
```

Directory of \texcours*.*

```
-----
CWI        3-28-94  9:55 by CWI
DOOB       4-18-94  15:11 by Doob
EIJKHOUT   4-18-94  15:24 by Eijkhout & Poppelier
HOENIG     4-18-94  15:14 by Hoenig
```

Directory of \texcours\horak*.*

```
-----
MFSOURCE   3-28-94  10:21
```

Directory of \texcours\salomon*.*

```
-----
FIGURES    4-18-94  18:16
TEXTS      4-18-94  18:16
```

Directory of \texhax*.*

```
-----
1986       4-18-94  13:37
1987       4-18-94  13:37
1988       4-18-94  13:37
1989       4-18-94  13:37
1990       4-18-94  13:36
1991       4-18-94  13:36
1992       4-18-94  13:36
1993       4-18-94  13:36
1994       4-18-94  13:36
IDX        4-18-94  13:37
TXH        4-18-94  13:36
```

Directory of \texshell*.*

```
-----
DOC         3-14-94  9:33 documentation
REMOVE     3-14-94  9:33 remove script
```

Directory of \texshell\doc*.*

```
-----
GERMAN     3-14-94  9:33 German documentation
```

Directory of \tex_nl*.*

```
-----
1989       4-15-94  15:54
1990       4-15-94  15:54
1991       4-15-94  15:54
1992       4-15-94  15:54
1993       4-15-94  15:54
1994       4-15-94  15:54
```

Directory of \tools*.*

```
-----
MKBAT      4-19-94  16:56 macro expander utility
```

Directory of \uktex*.*

```
-----
1987       4-15-94  15:55
1988       4-15-94  15:55
1989       4-15-94  15:55
1990       4-15-94  15:55
1991       4-15-94  15:55
1992       4-15-94  15:55
1993       4-15-94  15:55
1994       4-15-94  15:55
IDX        4-15-94  15:55
```

Occupied space on CD-ROM > 479 Mbytes
Total number of files > 18.890

Aankondiging cursus \LaTeX

Erik Frambach & Wietse Dol

Doelgroep

- zij die al (enige) ervaring hebben met \LaTeX , maar graag de *finesses* willen leren;
- zij die overwegen \LaTeX te gaan gebruiken, en een *overzicht van de mogelijkheden* willen krijgen;
- zij die \LaTeX als *systeembeheerder* (willen) ondersteunen en detailkennis van de implementatie willen opdoen.

Inhoud

de installatie:

- welke directory-structuur is handig/noodzakelijk
- hoe zitten de `.set` bestanden in elkaar
- welke onderdelen kun je missen, welke niet
- trouble shooting
- Frequently Asked Questions

de gebruikerskant:

- welke menu's zijn er, wat kun je er mee
- welke speciale functies kent de editor (Qedit en TSE)
- hoe maak je een eigen \TeX format file en Metafont format file
- plaatjes in \TeX documenten opnemen
- automatische fontgeneratie

onder de motorkap:

- hoe voeg je je eigen printer toe
- hoe voeg je een PostScript font toe
- hoe beperk je de mogelijkheden voor gebruikers
- hoe kun je het gebruik van \LaTeX meten
- hoe voeg je eigen functies en utilities toe
- 4DOS en zijn Batch mogelijkheden

Vorm

korte inleidingen en demonstraties, zelf proberen. Er staan 24 PC systemen ter beschikking, voor maximaal 48 personen. Bij voldoende belangstelling zal in groepen van voorkennis en interesse worden gewerkt.

Datum

10 juni 1994 : de dag na de NTG bijeenkomst

Docenten

Wietse Dol, Erik Frambach

Plaats

Rijksuniversiteit Groningen, Zernike-complex, Paddepoel, Rekencentrum

Kosten

Hfl. 75,-
overmaken op postgiro 1306238, t.n.v. penningmeester NTG, Zoetermeer, onder vermelding van 'cursus \LaTeX Groningen'.

Inschrijving

per e-mail naar 'cgl@risc1.rug.nl', of schriftelijk bij de NTG, Postbus 394, 1740 AJ Schagen, *onder vermelding van voorkennis en eventuele wensen* aangaande de cursus. Tegelijkertijd dient het cursusbedrag te worden overgemaakt.

Table of Contents TUGboat

Volume 14.4, and 15.1

December 1993 / March 1994

Abstract

TUGboat table of contents and related files are on `math.utah.edu` in `pub/tex/pub/tugboat`, also accessible via `tuglib@math.utah.edu` server by 'send index from tex/pub/tugboat'. The files can also be found on CTAN in directory `digests/tugboat/t-of-c`.

TUGboat 14.4 (December 1993)

- Addresses, p 369-370
- **Christina Thiele**
Opening words:
T_EX at meetings of other societies; Free-Net; Renew for 1994, p. 371–371
- **Barbara Beeton**
Editorial comments:
Reminder to potential TUGboat authors; Call for volunteers, p. 372–373
- TUGboat wish list, p. 373
- **Malcolm Clark**
N_EXI_EX: A personal view, p. 374–380
- **Philip Taylor**
NTS update, p. 381–382
- **Thomas Becker**
Two extensions to GNU Emacs that are useful when editing T_EX documents, p. 382–386
- **Donald E. Knuth**
Icons for T_EX and METAFONT, p. 387–389
- **Armin Liebl**
Bibview: A graphical user interface to BibT_EX, p. 390–395
- **Nelson Beebe**
Bibliography prettyprinting and syntax checking, p. 395–419
- **Kevin Carmody**
A tough table becomes easy with P_TC_TE_X, p. 420
- **Claudio Beccari**
Book Review: 'P. W. Abrahams, K. Berry and K. Hargreaves, T_EX per l'impaziente', p. 421–422
- **Helmer Aslaksen**
Ten T_EX tricks for the mathematician, p. 423–424
- **Victor Eijkhout**
The bag of tricks, p. 424
- **David Rhead**
The 'operational requirement' for support of bibliographies, p. 425–433
- **Richard Bland**
Relative moves in L_AT_EX pictures, p. 433–437
- Calendar, p. 438–439

- **Barbara Beeton**
Production notes, p. 439
- Coming next issue, p. 440
- Institutional members, p. 441–442
- TUG membership application, p. 443–444
- Index of advertisers, p. 442
- T_EX consulting and production services, p. 445

TUGboat 15.1 (March 1994)

- Addresses, p. 3-4
- **Christina Thiele**
Opening words:
Board news; The annual meeting; T_EX in new places; Announcing EuroT_EX'94; Office updates, p. 5-6
- **Barbara Beeton**
Editorial comments:
TUGboat wish list; Another award for DEK; New magazine on type and typography; Hyphenation and exceptions, p. 6-7
- **Donald Arseneau, Raymond Chen and Victor Eijkhout**
The T_EX hierarchy, p. 7-9
- **Claudio Beccari**
Typesetting of ancient languages, p. 9–16
- **Claudio Beccari**
Comments on the paper 'Typesetting Catalan texts with T_EX' (TUGboat 1, no. 4(3), pp. 252–259), p. 17
- **Gabriel Valiente Feruglio and Robert Fuster**
Comments on the comments: Typesetting Catalan texts with T_EX, p. 17–18
- **Nico Poppelier**
Book review: 'George Grätzer, Math into T_EX', p. 18–19
- **T. L. (Frank) Pappas**
Book review: 'Stephan von Bechtolsheim, T_EX in Practice', p. 19–21
- **Claudio Beccari**
Book review: 'Gianni Gilardi, Il T_EX-Introduzione al linguaggio e complementi avanzati', p. 21–22

- **Merry Obrecht Sawdey**
Book review: 'Erik Spiekermann & E. M. Ginger, Stop Stealing Sheep', p. 22–24
- **Nico Poppelier**
Book review: 'Eric van Herwijnen, Practical SGML', p. 24–25
- **Christine Detig and Joachim Schrod**
Book review: 'Donald E. Knuth, Literate Programming', p. 25–27
- **David Salomon**
Output routines: Examples and techniques, Part IV: Horizontal techniques, p. 28–40
- **David Salomon**
Verbatim copying and listing, p. 40–54
- **Victor Eijkhout**
The bag of tricks, p. 55–57
- **Hans van der Meer**
Random bit generator in T_EX, p. 57–58
- **David Salomon and Matthew N. Hendryx**
Slanted lines with controlled thickness, p. 59–62
- **Jonathan Fine**
On indexing and errors, p. 62
- **Chris Rowley**
L^AT_EX₂ ϵ update, dateline: 31 January 1994, p. 63
- **David Rhead**
Addenda: A suggested 'operational requirement' for L^AT_EX₃'s treatment of bibliographic references (TUGboat 1, no. 4(4), pp. 425–433), p. 64–65
- Calendar, p. 66–67
- TUG '94—Announcement and preliminary program (Santa Barbara, California, 31 July–4 August 1994), p. 68–69
- Call for papers, EuroT_EX '94 (Gdansk, Poland, 26–30 September 1994), p. 69–70
- **Barbara Beeton**
Production notes, p. 74
- Coming next issue, p. 75
- Meet the Board, Part I: Barbara Beeton; Jackie Damrau; Luzia Dietsche; Michael Doob; Michel Goossens; Tom Rokicki, p. 70–73
- Institutional members, p. 75–76
- TUG membership application, p. 77–78
- T_EX consulting and production services, p. 79
- Index of advertisers, p. 81