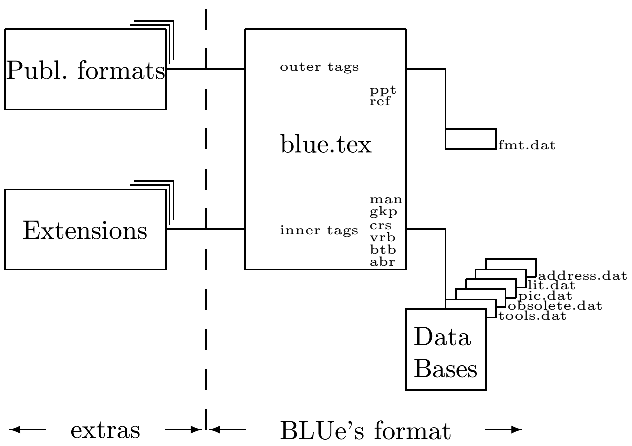
Publishing with TEX

—BLUe's Selection —

by

Kees van der Laan



Translated by Irina Gorbunova

Keywords: Bibliography, blue.tex, citations, compatible extension, cross-referencing, customization, databases, data integrity, education, index, letter, manmac, mark up, math, pictures, no-nonsense, plain TeX, programs, publishing, report, reuseable software parts, software engineering, suppprematism, tables, table of contents, table of examples, transparencies, verbatim.

ISBN: . . .

Classification: . . .

Date: February 1995

Acknowledgements

First of all

Don Knuth, Thank You!

for what you gave to the world, and to me in particular. I enjoyed so much studying your TeX works. I never perused a book so many times as *The TeXbook*.

I owe many thanks to numerous anonymous people who contributed to my (LA)TeX knowledge. I'm very grateful to NTG for bringing out early notes in MAPS, and to Erik Frambach and Wietse Dol for disseminating BLUe's Format system on the 4AllTeX CD-ROM. With pleasure I look back at BachoTeX 94, where among others Bogusłav Jackowski stimulated me to work on a manmac user's guide. This emerged in BLUe's Format system and fitted well with my ideas about a 'Publishing with you-name-it' series.

The macros and styles of the TeX Users Group, and the American Mathematical Society have been very inspiring. Of the former I borrowed the abbreviations and the approach of the OTR. The latter inspired me how to format references. I'm not aware to what extent the excellent user and installation guides of the AMS have influenced me. I like those for sure. Amy Hendrickson made me familiar with two-part macros. Amy's `\diagline` macro is part of blue.tex, although it is not much used by other macros, because of LaTeX's line fonts. David Salomon's courseware has been a wealthy source, especially for macro writing. Phil Taylor familiarized me with sequences of `\expandafter`s. His `\ignorewhitespace` has been incorporated in blue.tex. Bernd Raichle mentioned en-passant how to look at triples of `\noexpand`s. Joseph Romanovsky stressed the importance of memory management, which resulted in the general database approach. Frans Goddijn made me appreciate the frisky `\oldstyle` numbers. My perception of compuscripts has its roots in the SGML world. And, of course, I learned a lot from LaTeX.

Frank Poppe, Eric Veldhuyzen, and Jos Winnink, thank you for proofreading and for the useful comments. Peter Frambach polished my use of English.[1] Erik Frambach, again, was of great help in the proofing phase. Many a suggestion with respect to the layout was prompted by him. Irina Gorbunova, my host when attending CyrTUG 93, 94, translated the script into Russian.

I'm grateful to all of you.

 Kees van der Laan
Hunzeweg 57
9893 PB Garnwerd
The Netherlands
cgl@rc.service.rug.nl
31 5941 1525

[1] Any abuses left have to do with later versions he did not see.

Publishing with TEX

—BLUe's Selection —

by

Kees van der Laan

Abstract

The use of BLUe's format has been documented in a user's guide, to be released electronically in English and in Russian.

The functionalities provided are similar to those of LATEX, to mark up an article, a set of transparencies, a report, and a letter. The attitude is much different though. An example of the latter is processing of a modest index in one pass, completely within TEX.

The concept of two clusters of markup tags is introduced, called the outer level markup tags and the inner level markup tags. The former are responsible for the look-and-feel, and are kept to a minimum to cope with the inevitable change. The latter constitute the lower level issues, and are aimed at stability, invariance of representation and implementation, much in the spirit of the frozen kernel of TEX.

Many examples, exercises and answers have been included.

My basic attitude has been to make TEX simpler to use, to let it look less confusing. To select from the multitude of commands a handful, and to elucidate what we need most of the time.

New details treated are: the script concept, the separation of outer and inner level markup, using databases—formats, tools, references, pictures, and addresses—in a one-pass job with TEX, macros to format Pascal programs, a verbatim suite of macros to cooperate with AnyTEX, selective line-numbering of verbatims, a compatible extension of math cross-referencing, some matrix icons, a compatible extension of `\equalign` with multi-alignment points, a simple approach to formatting of partitioned matrices, a refined bordered table macro with the awareness of logical rows and columns and attributes to control layout, various codings of deterministic tables, a new look at manmac's `\samplebox`, creation of a modest index in one pass, design of a report format on top of blue.tex, and last but not least a uniform coding approach to start from two-part macros and to build one-part macros on top of those, parameterized by `\this`⟨*tag*⟩, `\every`⟨*tag*⟩, `\pre`⟨*tag*⟩, `\post`⟨*tag*⟩, and eventually `\`⟨*tag*⟩`name`.

I have built upon manmac and gkppic, and developed most of the macros myself. Ideas have been borrowed from the styles of TUG, AMS, from the proceedings of the EuroTEX and TUG meetings, and from the articles in the bulletins of the various user groups. Of course, many an idea came to mind when talking to friends at the various TEX User Group meetings. The markup of examples released earlier has been improved, simplified, or made more consistent.

This guide is about BLUe's Format, aimed at
formatting documents in English via plain TEX.

What do we mean by a translation?

For the time being it is just a translation of the
English into Russian, *not* a transliteration
of the contents.
The latter entails that an index in Russian
is *not* generated on the fly.
Another consequence is that this guide
is *not* about the use of CyrTEX.

Hopefully, the future will give birth to a real
Russian BLUe's Format project.

Irina Gorbunova
Kees van der Laan

Preface

Dear User,

It occurred to me that my detailed studies as published in NTG's MAPS, such as the BLUes series—Manmac, TUGboat, AMS—did not attract the attention it deserved, ;-))), let alone that people read it. They simply lack the time. Similarly, the BLUe's series enjoys a modest audience. Most TEXies are just busy using LATEX. Therefore, 'Publishing with TEX,' PWT for short, is not so much about reading, but has all to do with using.

My experience in macro writing comes from perusing Knuth's books, attending Amy's and David's advanced TEX classes, looking into the works of colleagues, and inevitably experimenting. The fundamental macro writing TEXniques I have assimilated will be reported in NTG's MAPS, in backside of the envelope contributions, in what I call the Paradigm series.

BLUe's format, its extensions and this user guide 'Publishing with TEX,' are shareware available,[2] and will be distributed electronically on diskette, via NTG's CD-ROM, or can be plucked from the CTAN. The relation between the various BLUe's articles and PWT is that the latter is a user's guide, while the former explain the coding too, are about the whys-and-hows, about insights and hindsights. However, remember that PWT is not an introduction to TEX nor a treatise on macro writing.

PWT has been available in February 1995, in English. A version in Russian is scheduled to be released before EuroTEX 95, and CyrTUG 95. For the latter I'm dependent on my host for my various stays in Moscow, Irina Gorbunova, who volunteered to translate the work.

My aim has been to make PWT as useful as possible, even for those who don't intend to use blue.tex. Numerous examples show how to mark up in TEX various typographic challenges known in the computer-assisted information processing and dissemination world. The above does not imply that the PWT script can't be printed. No, not at all, it can, and you will be surprised, hopefully, by the thin booklet of 125 odd pages, which will come off your selfpublishing press, some day.

I strove for a logically and physically thin system. The user's guide is physically thin and even thinner logically. If you are not interested in the math markup, just skip the chapter. Within a chapter examples can be skipped of course. The chapters are independent with 'thin' interrelations. The files, especially the databases, allow you to load only what you need.

The layout has been kept simple. The page width has been determined by my MAC screen. A large page length is there to reduce the number of pages and to prevent underfull vboxes. The primary medium is a file, not a book.

For future releases, I intend to include 'markup pearls' from readers. In the meantime, I'm pondering about hyperlinking the script to the TEXbook script, and acquainting myself with METAFONT and METAPOST. Another challenge, but fun. For the time-being, a printed version of PWT, with the TEXbook at your elbow, or with windows open on each of the files needed, will hopefully ease your TEXing task. If not, let me know... And for the rest

let us TEX with confidence

I hope that my approach to publishing with TEX—a result of some 7 years of TEX use and of some 20 years of programming numerical software—will be beneficial to new and old (LA)TEXies alike.

Kees van der Laan

[2] I would welcome $ 25 from each happy user, to facilitate investment in hardware to maintain the format. The PWT booklet alone is worth it :-). Of course, you will be added to my register.blu database. Do provide complete affiliation, email included. Users in Russia can just send me a nice mail with their affiliation for the time being.

Preface
Table of contents
Introduction
- Understanding Knuth's markup
- BLUe's format
  - The syntax
  - Variants
- The Model
- Notations and definitions
- Markup conventions adopted

Text
- Paragraphs
  - Parameterization
  - Special paragraphs
  - Customization
- Headings
- Use
  - Items
  - Centering
  - Quotations
  - Non-centered displays
  - Setting text apart
  - Pascal programs
  - Syntax diagrams
  - Verbatims
  - Exercises and answers
  - Footnotes

Mathematics
- What is the problem, Doc?
- The extras
- Use
  - Plain's display maths
  - Examples from the TeXbook

Tabular material
- Bordered table
  - Model
  - Markup tags
  - Customization
- Use
- Kaleidoscope
  - Font charts

Pictures
- The roots
- Using pictures
- Creating pictures
  - Samplebox diversions
  - Picture environment
  - Markup of environment
  - Line thickness
  - Picture objects
  - Curves
- Adding a picture to pic.dat
- Examples

Coping with References
- Using references
- Cross-referencing
- Adding a reference to lit.dat
- Customization

Creating an Index
- Use
- Markup of IR-s
  - Syntax
  - Markup
  - Spaces
- Special tokens
  - Sorting keys
- Ordering
- Typesetting the index
- Customization
  - Adding tokens to be ignored
  - Modifying ordering
  - Enriching the index
  - Typesetting the enriched file
  - Processes and files

Customization
- Changing look-and-feel
  - The page size
  - The header and footer
  - Section beginnings
- BLUe script into MAPS
- Other languages

Afterthoughts

Appendices
- A: Answers to the exercises
- B: Formats
  - Use
  - Concert
  - Letter
  - Report
  - Transparencies
- C: Tools
- D: BLUes, BLUe's and Paradigms
- E: List of Examples
- F: Main Tags
- G: Obsolete database
- H: History of Changes
- I: Index
- J: Table of Contents

Introduction

A plea is made for simplicity after having made the right a priori choices. The aim is to provide a user's guide such that blue.tex can be used with confidence. TEX and its benefits are explained in a nutshell. The contents of each chapter is summarized and the audience addressed. The model of BLUe's format is given. The concept, syntax and template of a BLUe script are touched upon. At the end a list of notations and definitions has been included.

Scientific tradition has it that simplicity can be attained once the right a priori choices have been made.

In relation to using TEX this means to get your priorities right, to keep in mind that it is all about what you have to say, to what you developed and wish to communicate.

Next to the a priori choices, selection is important. The 'handful' of tags needed in practice most of the time, at least by me, will be treated. TEX is just one of the tools needed for computer-assisted information exchange, and therefore there is no use in (sub)optimizing.

Understand BLUe's format sufficiently and use TEX with confidence!

The spirit of this work on using TEX is to guide a user towards preparing documents with as little knowledge of the nitty-gritty of TEX as possible. A tool should not become part of the problem.[3]

TEX—designed in the late seventies and frozen in the early nineties—is a tool for formatting documents by computer. Together with its twin sister METAFONT it has revolutionized computer-assisted typesetting. The benefits of TEX are

- quality
- availability on most if not all computers (in the Public Domain, turn-key)
- independence from input and ouput devices, and last but not least
- stability.

Because of this, an ASCII script with TEX markup is portable in place and time.

I'll show that TEX is suited for structural markup. This in contrast with the widespread belief that this is the virtue of LATEX. Not true, IMHO, with all respect. LATEX provided a user's guide along with its macros and emphasized the concept of separate style files.

From the ample TEX commands I'll select BLUe's subset for practical day-to-day use. To format an article—a set of transparencies, and some more—with blue.tex, that is what it is all about. These representations of a script are common. Write down the results, and support the aftertalks by transparencies.

The 'BLUe's format' section defines the structure of a BLUe script, and provides a template, to be filled in. How to mark up copy in detail, is subject of the chapters 'Text,' 'Mathematics,' 'Tables,' and 'Pictures.' How to add a list of references and how to make citations to that list is treated in the chapter 'References.' The chapter 'Index,' is about how to create and include an index. The chapter 'Customization' is about modifying blue.tex. Each chapter contains exercises. As Appendices I included 'Solution to the Exercises,' 'Formats,' 'Tools,' 'BLUes, BLUe's, and Paradigms,' 'List of Examples,[4]' 'Table of Tags,' 'History of

[3] My adage is 80% of the quality with 20% of the energy. Relevancy prevailed high-brow-ness. Common scripts marked up by sound, non-esoteric TEXniques.

[4] Although this work is not a cookbook, it's handy to have a quick access to the examples, because it has all to do with using, not in the least by example.

Changes,' 'Obsolete stuff,' 'Index,' and 'Table of Contents.' A similar approach as in 'Publiceren met LATEX' has been on my mind. Ideally, the material dealt with should have been the same, with only the LATEX tool exchanged for TEX. The underlying idea is that the functionalities required, and the essence of the exercises, should be independent of the tool used. The same setup and exercises should allow a treatise on you-name-it word processor.[5]

Floating aspects have been used in a special sense. Information is set in a box and inserted via a pasteup control sequence at your place of choice. Of course the usual inserts of plain for floats, `\topinsert`, `\midsert` closed by `\endinsert`, are still there to be used

I did not include a chapter on basic notions in publishing or book production, because that information can be found elsewhere, and would otherwise unduly increase the size of this work. I did not include a survey of 'TEX flavours and worlds' either, for similar reasons although it comes closer to the subject. For TEX flavours et cetera, consult the archives of the worldwide electronic networks to find for example 'What is TEX and METAFONT all about?'

Nothing of the world of PostScript coupled to TEX has been included, jeopardizing PostScript's scalability of fonts, its rotation capabilities, and its possibilities with colours. I take the fonts which come with TEX for granted, assuming English as the lingua franca for scientific communication. The latter implies that (multi-)lingual issues have not been touched upon either.

Font switching and font selection has been treated poorly, to put it mildly. For the moment I stay with Knuth and copied his size-switching macros. BLUe's format contains the size-switching macros `\Large`, $\approx$ 17pt, used for chapter headings, `\large`, $\approx$ 14.5pt, used for transparencies, and `\small`, i.e., 8pt, used for footnotes. Next to these `\tenpoint`, `\ninepoint`, and `\eightpoint` have been inherited from manmac. However, the abstraction of mapping the linear space of fonts onto a logical 5 dimensional space, which is the basic idea in NFFS, is nice and might become very useful some day.

Because of these a priori choices this guide remained small and will hopefully be perceived as simple.

The audience in mind consists of scientific workers, especially the growing breed of self-publishing authors, with substantially complex copy, especially complex math and tables. Authors who concentrate on the contents welcome a reasonable print-out, exchange their works via electronic networks, submit (LA)TEX scripts to publishers, like to understand their tool, and last but not least aim to use it confidently with minimal markup.[6] In netland one can't rely much on publishers for the moment, one is on one's own. There is even the danger of becoming a TEXhacker after some years, because TEX will serve your lifetime.

For running a BLUe script with text not much TEX knowledge is required. When details of math, tables, and graphics come around the corner, familiarity with David Salomon's courseware—MAPS special 1992, also available on NTG's CD-ROM—is assumed.

The key to understanding Knuth's markup

It did take me quite a while to understand Knuth's markup, :-))). Biased by my programming experience, and influenced by SGML and ... LATEX, I was on the wrong track. My mind was full of block structures with rigorous opening and closing tags, and parameterization via arguments of procedures. I was blocked for the simple approach of *text replacement*, and had to unlearn much, in order to become innocent again, to open my mind for Knuth's approach.

ASCII text with visual layout
and
minimal markup tags added

The royal road to understanding and appreciating Knuth's markup is starting from ASCII text, what you would write by hand. The next step is inserting markup tags, as few as possible, that is, enriching the ASCII file with minimal markup.[7]

---

[5] The main difference with the early work on LATEX is that now the markup of a BLUe script is the issue.

[6] TEX markup is extremely tolerant. Many ways are possible, opening the door to quick-and-dirty markup. A discipline is needed.

[7] It is quite another task to implement those minimal tags, however.

An example? Undo the examples in the Example Formats chapter of *The TEXbook* of the markup, and insert your own markup, and you will know what I'm talking about. I hope you will come to the same conclusion as I did

Knuth's markup is unsurpassed!

I'll come back to the matter in the chapter 'Text' and in the Appendix 'Formats,' especially with the letter format.

BLUe's format

The format is a system to format compuscripts. It is a compatible extension of plain TEX and organized as a 'kernel + modules,' similar to the setup of FORTRANxx. It is designed to be supportive, to suit your needs. You are Ben Lee User. It is not imposing.[8]

The concept of a BLUe script is at the heart. blue.tex allows one or more scripts to be processed[9] in one run, even non-BLUe scripts. As an example of the latter

Thank you. `\bye`

runs with blue.tex. This tolerance is handy while proofing parts.

Example *(BLUe script)*

Generally, I use a template, an empty script as a fill-in form, with the main markup tags already there.[10]

```
%Template BLUe script---outer level markup
                                   %Preliminary part, with
                                   %elements in arbitrary order
\issue{MAPS 94.2}
\title{BLUe's Format}
\subtitle{The best of both worlds}
\keywords{...}
\abstract{...}
\contents{Introduction
  \se Why?                         %\se acronym for subelement
  ...
  Acknowledgements, Conclusion, References}
\references{\⟨name1⟩...\⟨namen⟩}
\pictures{\⟨name1⟩ pic...\⟨namen⟩ pic}
\acknowledgements{...}
\beginscript                       %Marked up copy proper comes next
\head{Introduction}
                                   %Various parts with inner markup
...
                                   %Ended by the following back matter
\pasteupacknowledgements
\pasteupreferences
\endscript
```

Once used to the template, the outer markup is a trifle. Of course, in order to process the script it should be preceded by `\input blue.tex`. On my Mac Classic II blue.tex is a format. One could even think of a bluetex command similar to LATEX.

---

8 I know of the defensive attitude of publishers. They welcome the benefits of imposing formats to prevent the use of macros which hinder production. There is much truth in there, given the reality of the complex—or better, the unusual—TEX markup language. However, there is no use in protecting an author against himself. Simple supporting tools should make the imposing attitude superfluous.

9 Either in sequence or nested. The former is handy when several contributions have to be run in one job, for a journal for example. The latter has been used in this script to switch temporarily to `\letter`, casu quo `\transparencies`.

10 The order does not matter much for the elements before `\beginscript`. They can be supplied in any order. Because of the short markup tags the feature to let the computer prompt the `\end`⟨*tag*⟩, once the `\begin`⟨*tag*⟩ has been typed in, is not so relevant.

1.1 How to override the defaults for `\author` and the like?

1.2 How can we process more scripts in one run?

Default the look-and-feel in print resembles much *TUGboat*'s (two-column) layout.

The syntax for a BLUe script reads as follows.[11]

⟨BLUe script⟩ ⟶ `\input` blue.tex ⟨scripts⟩
⟨scripts⟩ ⟶ ⟨script⟩ | ⟨script⟩⟨scripts⟩
⟨script⟩ ⟶ ⟨preliminary⟩⟨scriptoptions⟩`\beginscript`⟨sections⟩⟨indexing⟩`\endscript`

with refinements

⟨preliminary⟩ ⟶ ⟨index⟩⟨format⟩⟨language⟩`\title`⟨infopart⟩
⟨index⟩ ⟶ ⟨empty⟩ | `\loadindexmacros`
⟨format⟩ ⟶ ⟨empty⟩ | `\concert` | `\letter` | `\report` | `\transparencies` | \⟨*yourformat*⟩
⟨language⟩ ⟶ ⟨default⟩ | `\english` | `\russian`[12] | \⟨*yourlanguage*⟩
⟨infopart⟩ ⟶ ⟨empty⟩ | ⟨infoelement⟩⟨infopart⟩
⟨infoelement⟩ ⟶ `\subtitle` | `\issue` | `\input` | ⟨TeXcode⟩
| `\author` | `\address` | `\netaddress`[13]
| `\keywords` | `\abstract` | `\contents` | `\references` | `\pictures`
| `\acknowledgements`
⟨scriptoptions⟩ ⟶ ⟨empty⟩ | `\everyscript` | `\thisscript` | `\everyscript\thisscript`

The sections consist of the tree of head structures with IM copy, inner marked up copy proper.

⟨sections⟩ ⟶ ⟨empty⟩ | ⟨section⟩⟨sections⟩
⟨section⟩ ⟶ ⟨head⟩⟨paragraphs⟩
⟨head⟩ ⟶ `\head` | `\bluesection`[14]
⟨paragraphs⟩ ⟶ ⟨empty⟩ | ⟨paragraph⟩⟨paragraphs⟩
⟨paragraph⟩ ⟶ ⟨subsections⟩ | ⟨IM copy⟩
⟨indexing⟩ ⟶ ⟨empty⟩ | `\sortindex\pasteupindex`

The above can be repeated for subsections and subsubsections.

Variants Via a control sequence—called an attribute in the SGML world—the format used can be changed, resulting in a different look-and-feel in print. For example, via `\onecolumn` within the default format. This is meant for a script in total. However, two-column and one-column scripts can be processed in one run.

The control sequence `\transparencies` changes the result into a representation suited for copying on transparencies. Adaptation of the script is mainly about deleting.

Especially for PWT I designed the report format. It is essentially one-column, and a mod on top of the default. The control sequence `\report` yields the report layout. The path 'From-report-to-transparencies' comes down to stripping off the superfluous. See the Appendix 'Formats' for details.

I also included Knuth's letter format, `\letter`,[15] and concert format, `\concert`.

---

[11] I used the same Backus-Naur formalism for the syntax diagrams, as adopted by Knuth and explained in *The TEXbook* 268, 269.
[12] In progress as spinoff of the translation effort.
[13] Defaults.
[14] A minimal markup alias, a tribute to manmac.
[15] Well, I updated Knuth's approach a little, if I'm allowed to say so. Important is to provide for mail merge and in general to let the format cooperate with an address database. And of course I simplified the macros a little. No different modes of text processing.

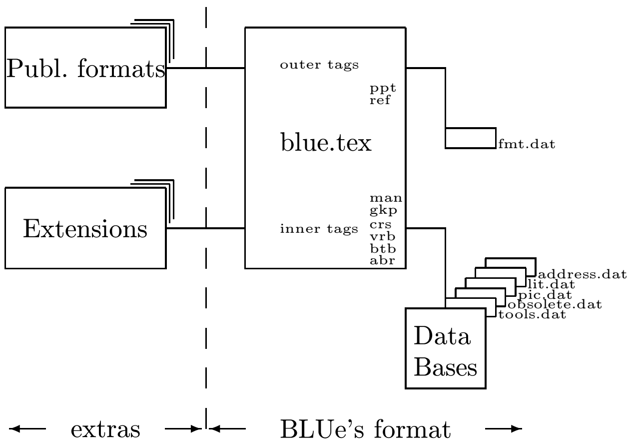
Extras The Appendix 'Tools' is mainly about the lower level extras. It provides a method to organize all your macros. It is not part of blue.tex, but can cooperate smoothly without overhead, respecting the 'little user,' i.e., the user with modest demands.

It is also important to think of a garbage bin. Somewhere to keep your obsolete stuff, macros or formats, in order that your scripts can still be formatted without change in future. For that purpose I introduced the obsolete.dat database.

Finally, I coupled my address.dat database, that is letters can find addresses in that database, similar to a bibliography which will take the references from a bibliography database.

The model

BLUe's format is a personalized formatting system. It's yours! It knows about your context. It is based on manmac, gkppic, and the use of databases.



What does the model say? One way of looking at it is to think of inner and outer worlds.[16] The extras part reflects the outer world in general. Your publisher and his tools, casu quo formats, and your colleagues who provide macros for general use. The BLUe's format part—your inner world—divides the markup tags into two clusters, supported by databases.[17] These databases not only support the handling of the usual collection of references. It also provides the mechanism for elegantly storing pictures. Even newer is using the database approach to manipulate formats and tools. We don't need all those separate files any longer. This approach can also clean up the organization of the multitude of font files, I presume.

Another way of looking at the model is that the model provides a structure for keeping pace with change and progress.

You have to do the work. Polish what you have to say. Mark up, as naturally as possible, by using formatting goodies from the past like manmac and picmac. I like minimal markup, that is

supprematism in markup

in the spirit of Malevich's 'White cross on a white background,' let us say.

Build up your collection of works relatively independently of context.[18] Communicate what you have lived through, and keep in touch with developments.

When you are a self-publishing author, that is all there is. If not, you have to obey the wishes of your institute or publisher, meaning that outer level issues have to be adapted. Outer level commands govern the title part and the global page layout, like headings of

---

[16] Knuth had the outer-inner duality on his mind already, as can be deduced from his `\outer \def`-s.

[17] Database in this context means essentially a multi-file file. Each tool or macro set is part of a file. I prefer to have separate clusters for formats, tools, references pictures and addresses. The global structure has been simplified. What is needed will be loaded selectively. The best of both worlds.

[18] This looks trivial but experience has it that early *TUGboat* articles, for example, can't be formatted without adaptation, because the formats used are obsolete. An intrinsic question is whether we should update or republish old work. Up till now I have updated my works. All my (LA)TEX articles run with blue.tex. When composing this work, I decided not to update old works any longer. That is history from now on. This work contains the improvements, if any :-).

sections, running headers and footers, the font(s) used, the number of columns, the page size, paragraph indentation and separation, and the like.

The aim of the inner level set of markup tags is to remain invariant, with the purpose that these tags can be used in any context now and in future.

The borderline between inner and outer markup is context-dependent, and therefore undefined. I like to think in terms of overlapping clusters. What you might call outer I might call inner, and vice versa, not to mention your publisher's opinion. A pragmatic description of the group of your outer commands are those which you have to change when you submit your work to your publisher or a second publisher.

The wish to modify has been anticipated in BLUe's format. For that purpose either control sequences, which represent a different format, can be invoked, or a handful of outer level markup commands, casu quo parameters, can be customized.

Also the open-endedness is there. At the higher level the number of mappings from BLUe's format into publisher formats can grow. At the lower level you can add macros. And what about the old stuff? The old versions can be stored in the obsolete.dat database, with the purpose that BLUe scripts can still be formatted in future, without change.[19] And if you have a database of your own, just make the connection.

Portability and reusability are served by writing macros in plain TEX, the lowest common TEX language of all the flavours. The multitude of tools have been organized in a database to load selectively from.

The reusability of formats, tools, references, pictures, and addresses has been eased by storing these in databases. By this way I obeyed the Samelson principle, which comes down to

Don't pay for what you don't use.

Notations and definitions

address.dat is the address database to cooperate smoothly with `\letter`.

AMS-TEX denotes the (LA)TEX system in use by the American Mathematical Society.

ASCII is an acronym for American Standard Code for Information Interchange.

BLU is Knuth's nickname for Ben Lee User. I adopted his cousin BLUe.

BLUe script is an ASCII file with TEX markup inserted, to be processed by blue.tex.

BLUe's format is the collection of the formatting tools: blue.tex file (the kernel), and the databases fmt.dat, tools.dat, address.dat, lit.dat, pic.dat (modules).

blue.tex denotes the kernel format to typeset scripts.

CTAN is the acronym for Comprehensive TEX Archive Network. The resource of (LA)TEX materials on the internet. It comprises various sites with their contents automatically mirrored.

Customization occurs a lot in this work. On the one hand it is explained what to do, and on the other hand it is indicated how to customize the macros.

gkppic is the acronym for the macros used by Graham, Knuth and Pastashnik to format 'Concrete Mathematics.'

fmt.dat denotes the database of formats to be used along with blue.tex.

LATEX is Lamport's system on top of TEX to format articles, reports, letters, books and so on.

lit.dat is the literature database to cooperate smoothly with blue.tex.

IMHO stands for In My Honest Opinion, usually with all respect.

manmac stands for the macros used by Knuth to format *The TEXbook*.

Markup stands for the insertion of instructions in a script. There are various markup notions.

Automatic markup is used by me when the computer generates the markup.

Descriptive markup abstracts from concrete printers and emphasizes the descriptional nature. Agreed a little vague :-))).

[19] Well, . . . let us hope little change.

Functional markup emphasizes the functional character of the elements.

Inner markup tags is the set of markup tags that remains invariant when processing a script with another TEX flavour.

Logical markup is all about the logics of the script.

Minimal markup stands for markup with as few as possible markup tags.

Outer markup tags is the set of markup tags which needs to be customized when processing a script with another TEX flavour.

Recursive markup is used by me especially when the markup is generated by the computer by means of recursion.

Structural markup emphasizes the structure of the script.

Visual markup denotes the implicit markup by white space.

I'm not claiming that these descriptions enjoy general acceptance; no, it helps to indicate the various backgrounds and intentions, as I have perceived them and hinted at in this script. IMHO, there is a lot of overlap, and no generally accepted list of definitions.

Miniature is an initial of a chapter. Historically, a nice painting. In compuscripts generally a character from a special font.

NFSS is the acronym for New Font Selection Scheme. Its purpose is to describe the selection of fonts logically. A mapping of the 1 dimensional space of fonts onto the logical 5 dimensional space as introduced by NFSS, with dimensions encoding scheme, family, series, shape, and size.

OTR acronym for OutpuT Routine.

PDF acronym for Adobe's Portable Document Format.

pic.dat is the picture database to cooperate smoothly with blue.tex.

`\PiCTeX` a system to format pictures with plain TEX, due to Wichura.

Parameterization denotes ways to parameterize a macro.

PostScript denotes Adobe's—de facto standard—page description language.

References are stored in lit.dat, which contain even the works I don't remember explicitly but have built upon undoubtedly.

Script is my term of a file with (TEX) markup.

Section is a document element, that is, a head followed by paragraphs.[20]

SGML Standardized Generalized Markup Language. The ISO standard activity for markup, ISO8879.

Sorting keys are pairs consisting of a control sequences and the replacement for sorting the item in the index.

Syntax for tags: `\begin`⟨*tag*⟩, `\end`⟨*tag*⟩, `\`⟨*tag*⟩; `\pre`⟨*tag*⟩, `\post`⟨*tag*⟩ placement within context; `\this`⟨*tag*⟩, `\every`⟨*tag*⟩ handling of options

tools.dat is the database of tools to cooperate smoothly with blue.tex.

Markup conventions adopted

The preliminary pages are treated separately, don't take numbering, and have a label in the margin.

The chapter headings are set in a larger type. The title is used in the headline (without number), the list of examples, the table of contents, The (hidden) number of the chapter has been used in the exercises, the table of contents, and the list of examples. Chapter headings and section headings are centered and don't take a number. The appendices headings have their number printed as a capital letter.

Headings don't take a period. Run-in text can be recognized from the context.

Headlines have the title of the chapter left and the title of the report right. They are both set in slanted type.

[20] Analogous for `\subsection`and `\subsubsection`.

\parindent has been set to 0. Because of this the footnote numbers hang out. A \generalindent has been used.

Exercises take a number, no further indication has been used.

Examples are not numbered in the text, but labeled by the word 'Example' and take a label to characterize the example. The label is reused in the list of examples, next to the hidden number. Punctuation marks and footnote markers are used in such a way that the footnote marker appears after the punctuation mark. (Except for question marks and exclamations.) Quotation closings come after that.

Now and then an explicit \newpage has been inserted to prevent underfull vbox messages. No floats have been used, in order not to disturb the sequential ordering.

Copyrights BLUe's Format system

The software

blue.tex

fmt.dat

tools.dat

is shareware available. It is not allowed to make any profit on it. A happy user is requested to send US$25 to the author (twenty-five US dollar)

The contributors will be added my register.blu database. Russian users are relieved for the moment from the shareware request, with respect to personal use. Auxiliary and personalized files are the databases

address.dat	addresses
lit.dat	references
pic.dat	pictures.

The main reason to ask for shareware support is to protect myself against too many request, and to raise some funds to continue my TEX work.

The user's guide 'Publishing with TEX,' and the accompanying articles are free for personal use.

March, 1995

Kees van der Laan
Hunzeweg 57
9893 PB Garnwerd
The Netherlands
cgl@rc.service.rug.nl
31 5941 1525

Text

An introduction to the markup of text has been given. BLUe's format structure of text is nested to three levels, each preceded by a title. The building block is a paragraph. The markup for TEX's standard paragraph shape, the markup for the headings and the markup for some special text elements have been given. Examples of the latter are: items, quotations, display text, centering, text to be set apart preceded and followed by lines, PASCAL programs, verbatims, exercises and answers, and footnotes.

Introduction

Let us assume that we have ordinary text, to be typeset by computer. What to do? Let us begin by the beginning and write it down.

```
An unexpected party

In a hole in the ground there lived a hobbit.
Not a nasty, dirty, wet hole filled with the ends
of worms and an oozy smell, nor yet a dry, bare
sandy hole with nothing in it to sit down on or
to eat:
it was a hobbit-hole and that means comfort.

It had a perfectly round door like a porthole,
painted green,...
```

We must instruct the computer—that is, supply markup—to make something nice out of it. How?

As far as I understand it, Knuth's markup comes down to just preceding the title by `\beginchapter`?!? That is all. He makes use of the already available visual markup of blank lines. These blank lines separate document parts, be it a title from the text or paragraphs from each other.

Is that all? Essentially, yes. But—there is always a but—the role which `\beginchapter` has to play is not simple. Happily, that is not our concern. That is the job of the book designer in cooperation with the programmer. I take for granted the niceties of TEX's text processing—and which a regular user tends to forget—such as automatic handling of ligatures, justification and hyphenation over paragraphs, and ignoring of superfluous spaces in text. All this is not necessarily restricted to English.

After this oversimplification let us go back to BLUe scripts, full of the right spirit.

BLUe script

BLUe's format allows text in print to be nested up to 3 levels deep. The layout of the titles and the shapes of the paragraphs contribute much to the look-and-feel. The nesting is reflected in the markup tags `\head`, `\subhead` and `\subsubhead`. Also `\title` and `\subtitle` are there. The tag `\author` has been filled-in by default, meaning that you don't have to pay attention to it. It's *your* format. You are Ben Lee User.

Headings

As a tribute to manmac, BLUe's format provides the following outer level markup tags as an extra user-interface on top of the rigorous two-part macros.

`\blue(sub)title` ⟨*(sub)titletext*⟩

`\bluekeywords` ⟨*listofkeywords*⟩

`\blue(sub(sub))section` ⟨*(sub(sub))sectionheadtext*⟩

`\blueexample` ⟨*exampleheadtext*⟩

A handful, but—there is always a but—the processing on the fly has been lost. Note that all have a blank line—implicit markup—or `\par` as end separator. Reread page 202 of *The TEXbook*, please. I used them in this script, because they counter the curly braces mania, or to put it in Knuth's words

'…you needn't always enclose arguments in braces.'

The regular markup for headings reads as follows.[21]

`\head{`⟨*headtext*⟩`}`

Similar for `\subhead` and `\subsubhead`. No terminating periods are automatically typeset.

Parameterization The text of a heading is preceded by `\prehead` and followed by `\posthead`. Again, similarly for `\subhead` and `\subsubhead`. By default the placement within context—that is, the tokens contained in `\pre`⟨*tag*⟩ and `\post`⟨*tag*⟩—is determined by the following, apart from (stretchable) glue.[22] The latter fills up the page by a page break to prevent an underfull `\vbox` message.

\<*tag*>	\pre<*tag*>	\post<*tag*>
`\head`	`\bigskip`	`\nobreak\medskip\noindent`
`\sh`	`\medbreak\noindent\shindent`	`\quad`
`\ssh`	`\smallbreak\noindent\sshindent`	`\quad`

*sh stands for subhead, ssh for subsubhead

With defaults `\subheadindent0pt` and `\subsubheadindent\generalindent`. I used to mark up headings by the short form, i.e., by `\head{...}` instead of `\beginhead`…`\endhead`. From now on—back to the roots—I will use manmac's minimal forms. In headings I don't need much processing on the fly.

Automatic numbering is provided in the `\report` format. The idea is to use `\`⟨*tag*⟩`numbering` token variables.

The above is only part of the story, just the top of the ice-mountain. In a document we also like to reuse the head titles for table of contents, the index, and undoubtedly we need the head numbers sooner or later, whether they are hidden or not. So on the one hand we like to process a title on the fly, and on the other hand we like to store head information, as general as possible, for reuse. In the `\report` format I had to face the music and deal with those issues. Happily, the blue.tex kernel could do without it.

[21] Of course the long forms `\begin`⟨*tag*⟩ and `\end`⟨*tag*⟩ can be used too.

[22] See for those *The TEXbook* 355, especially `\beginsection`. 'The macro first emits glue and penalties designed to start a new page if the present page is nearly full, …' This mechanism has been borrowed for the various `\pre(sub(sub))head`-s.

Paragraphs

TEX's building block is the paragraph. This unit is split up into lines to be mapped on pages. The markup for a paragraph is minimal, i.e., it is generally implicit and near to natural. An occasional user is hardly aware of the markup. In most cases

a paragraph starts by the first letter, and
ends by a blank line.

The blank line is substituted by `\par` automatically. Explicit markup for a paragraph reads

```
\indent  %starts a paragraph, an empty box (size \parindent) is inserted
         %or \noindent, which just starts a paragraph
⟨paragraph⟩
\par     %ends the paragraph, and resets defaults of par shape pars
         %\hangindent and \hangafter
```

On encountering a vertical command in a paragraph, like `\smallskip` and its ilks, TEX automatically inserts `\par`, and because of this the paragraph is ended. Each script, and this one is no exception, contains a lot of ordinary paragraphs.

Parameterization The parameters which govern the standard paragraph shape are given in the following model.[23]

```
              \parskip
           \(base)lineskip
\parindentXXXXXXXXXXXXXXXXX
            \clubpenalty
           \(base)lineskip
XXXXXXXXXXXXXXXXXXXXXXXXX
         \interlinepenalty
           \(base)lineskip
                 ⋮
XXXXXXXXXXXXXXXXXXXXXXXXX
            \widowpenalty
           \(base)lineskip
XXXXXXXXXXXXXXXX\parfillskip
⟵             \hsize            ⟶
```

The left margin is `\hoffset` from the left. The distance from the left side of the paragraph to the left margin is the glue `\leftskip`. At the right side the glue has the name `\rightskip`. Between words the glue is `\spaceskip` and between sentences `\xspaceskip`.

The contents of the token variable `\everypar` is inserted before the paragraph text.

2.1 How can the parameters be set such that the last line of a paragraph is centered[24]?

2.2 One of the problems induced by paragraphs as building blocks is that it is not easy to provide for line numbering. How can we achieve this[25]?

Special paragraphs TEX provides `\parshape`, `\hangindent` and `\hangafter` for special paragraph shapes. A captivating example is provided in *The TEXbook* 101.

Example *(Miniature)*

History has it to start the first paragraph of a chapter with a 'Miniature.' Knuth in his 'Preface' showed how to mark up such kinds of paragraphs. This is an illustration of the use of `\hangindent`[26] and `\hangafter`, as well as of the use of `\smash` outside math mode. There is no need for a special 'dropped capital' macro, because it is really simple.

Remark. Realize that the reference point for the `\smash` is where the E starts, and that the first (2) lines are indented. Within the `\smash` you can move around as you wish, leaving the reference point and the other things outside invariant. Note the use of `\smc` to smooth

---

[23] Courtesy David Salomon. (A white lie, any migrated vertical material has been omitted.)
[24] Courtesy Anne Brüggeman-Klein.
[25] Courtesy Jan Kybic.
[26] `\hang` assigns the value of `\parindent` to `\hangindent`.

the transition from the miniature to the rest of the text. `\cglpic` is the minitiature used from the picture database pic.dat. For an nice application[27] see the miniature in the NTG information, on the inside back cover.

```
\pictures{\gpic}%load gpic
\parindent=24pt
\noindent\hang\hangafter-2
\smash{\lower13pt\hbox to0pt{%
\hskip-\hangindent\gpic
\hfil}}{\smc entle Reader}:
This is a handbook about \TeX,
a new typesetting system
intended for the creation of
beautiful books\Dash and
especially for books that
containa lot of mathematics.
```

GENTLE READER: This is a handbook about TEX, a new typesetting system intended for the creation of beautiful books—and especially for books that contain a lot of mathematics.

2.3 Compare the above simplified version of Knuth's opening sentence with the real one in the *The TEXbook* script.

2.4 At the end of each paragraph TEX resets the paragraph shape parameters. How can we retain these values[28]?

2.5 Is the scope of `\leftskip` and `\rightskip` restricted to a paragraph?

Example *(Flushed right paragraphs)*

At the end of each chapter in *The TEXbook* quotations are set as flushed paragraphs. A little simplified the following will do the job. The key issue is the use of `\leftskip` and the zero value of `\rightskip` and `\parfillskip`. `\manmacauthor` is the `\author` macro from manmac to set the author part in those quotations.

```
\leftskip0pt plus40pc
             minus\parindent
\rightskip0pt\parfillskip0pt
\everypar{\sl}\obeylines
I got the BLUes,
but I like it
\manmacauthor{Anonymous}
             (20th Century)
\bigskip
It contains all
\manmacauthor{DeK}(1978) \TB
```

I got the BLUes,
but I like it
— Anonymous (20th Century)

It contains all
— DeK (1978) *The TEXbook*

2.6 Why is `\everypar{\sl}` needed, instead of just `\sl`[29]?

Customization The hook for customization of paragraphs is the token variable `\everypar`. A nice application of `\everypar`, next to `\sl` above, is to number lines automatically in verbatim texts.

Use of special text elements

Examples of special text elements are: items, labeled by a bullet, by a number, or alphabetically; centering, quotations; (non-centered) displays; text to be set apart, preceded and followed by a line; PASCAL text; syntax diagrams; verbatims; exercises and answers; and footnotes.

The markup for these elements will be shown by examples.

[27] Courtesy Huub Mulders.
[28] Courtesy Marek Ryćko and Bogusł av Jackowski.
[29] Courtesy Staszek Wawrykiewicz

Example *(Items)*

At the basis are plain's one-parameter `\item` and `\itemitem`, which take as argument the item marker. The bulleted item, `\bitem`, comes from *The TeXbook* ex14.20. `\nitem`, `\aitem`, and `\Aitem` are similar extensions. The extensions provide the marker automatically, no argument is needed. Note that the closing tag is `\smallbreak`.

```
\bitem first item
\itemitem{--} subitem
\bitem second item
\smallbreak
```

- first item
  – subitem
- second item

Numbered items, instead of bulleted items, are obtained via the markup tag `\nitem` instead of `\item`. Alphabetically labeled items are obtained via `\aitem`, for lowercase item labels, and via `\Aitem`, for uppercase item labels.

Example *(Quotations)*

A special case of centering a block is the so-called quotation, indented on the left and the right by the same (fixed) amount, usually the value of `\parindent`.

```
pretext
\beginquote
Some text to be set apart,
indented on the left and
on the right
\endquote
posttext
```

pretext

> Some text to be set apart, indented on the left and on the right.

posttext

Add a little to this to obtain `\beginsummary` and `\endsummary`, abstracting in the markup from formatting details. A step towards descriptive or functional markup. The summaries at the beginning of each chapter of PWT have been marked up in this way. Perhaps it is also good to mention here that accents don't start horizontal mode. If you like to quote then start with `\leavevmode`—to enforce horizontal mode—followed by `\llap‘` or so.

Example *(Centering in display)*

Plain TeX provides `\centerline`. BLUe's format provides the extension `\displaycenterline`, to center a line with display glue for and aft, and `\begincenter` ... `\endcenter`, and `\begincenterblock` ... `\endcenterblock` to center each line of the block.

```
pretext
\displaycenterline{I $\heartsuit$
                          \TeX}
text in between
\begincenterblock
Sign of times?
The extraordinary rate of change.
\endcenterblock
intermediate text
\begincenter
TLC
tender loving care
that is all there is
\endcenter
posttext
```

pretext

$$\text{I } \heartsuit \text{ \TeX}$$

text in between

Sign of times?
The extraordinary rate of change.

intermediate text

TLC
tender loving care
that is all there is

posttext

Remark. Agreed, `\displaycenterline` is a special case and therefore superfluous.

2.7 Do `\begincenter...\endcenter` and `\begincenterblock...\endcenterblock` allow for page breaks?

Example *(Non-centered displays: one-line, multi-lines and multi-columns)*

`\begindisplay` and `\enddisplay`, *The TEXbook* 434–439, yields an indented display. Very powerful and user-friendly.

```
pretext
%TB 3
\begindisplay
   ``I understand.''
\enddisplay
posttext
```

pretext

"I understand."

posttext

Multi-line displays can be handled by ending each line in the environment by the control sequence `\cr`.The indentation is the value of `\displayindent`, default 0pt, next to the `\hbox to\parindent{}` because of the `\indent` in the (hidden) template.

```
pretext
%TB 4
\begindisplay
   a hyphen (-);\cr
   an en-dash (--);\cr
   an em-dash (---);\cr
   a minus sign ($-$).\cr
\enddisplay
posttext
```

pretext

a hyphen (-);
an en-dash (–);
an em-dash (—);
a minus sign ($-$).

posttext

Another subtle point is that more-column copy can be displayed, because a repetitive template has been used.

```
pretext
%TB 29
\begindisplay \hbadness10000
 \hbox spread-.666667em{The
 badness of this line is 100.}&
           (very tight)\cr
 \hbox spread-.333333em{The
 badness of this line is 12.}&
      (somewhat tight)\cr
 ...
\enddisplay
posttext
```

pretext

The badness of this line is 100.	(very tight)
The badness of this line is 12.	(somewhat tight)
...	

posttext

2.8 How can we center a `\begindisplay`...`\enddisplay`?

Example *(Setting text apart)*

`\beginlines` and `\endlines`, *The TEXbook* 583–591, is in display (zero-indented), which obeys the line structure and inserts a `\hrule` for and aft. For an example of use I borrowed the following from *The TEXbook*, where it is used in combination with in-line verbatim. `\obeylines` is on.[30]

```
%TB 343
\beginlines
|%This is the plain TeX format |
|%that's described in The TeXbook.|
\dots
\endlines
```

```
%This is the plain TeX format
%that's described in The TeXbook.
...
```

---

[30] Variant `\endlines` are `\weakendlines`, `\finalendlines`. The former uses a `\medskip` instead of a `\medbreak`.

Example *(Syntax diagrams)*

Markup tags are

`\beginsyntax...\endsyntax`	the environment
`\alt`	to start an alternative
`\is`	to define left element
`<` and `>`	to start and end a unit.

When a line starts with `<` it is assumed that a new rule starts. `\obeylines` is on. The end-of-line is redefined to take the appropriate action. In the example below a `\quad` is used in the markup to prevent starting a new rule. The % sign at the end of a line annihilates the effect of the e-o-l, and prevents a new rule to start.

```
%TB 268
\beginsyntax
<unitofmeasure>\is
\quad<optionalspaces>%
       <internalunit>
\alt<optional {\tt true}>%
          <physicalunit>
\endsyntax
```

⟨unit of measure⟩ $\longrightarrow$
    ⟨optionalspaces⟩⟨internalunit⟩
  | ⟨optional `true`⟩⟨physicalunit⟩

Example *(Programs)*

Program texts are special, due to the use of non-monospaced fonts. blue.tex assumes that the PASCAL texts have been prepared in such a way that they are insensitive to font spacing.[31]

```
\beginpascal
{Print length of third side of
 triangle given two sides and
 enclosed angle}
program EX4A(input, output);
var a, b, c angle : real;
begin read(a, b, angle);
   c:=sqrt(sqr(a)+sqr(b)-
                 2*a*b*cos(angle));
   writeln('The third side is', c)
end.
\endpascal
```

{ Print length of third side of
 triangle given two sides and
 enclosed angle }
program *EX4A(input, output)*;
var *a, b, c, angle* : *real*;
begin *read(a, b, angle)*;
    $c := sqrt(sqr(a) + sqr(b) -$
        $2 * a * b * cos(angle))$;
    writeln('The third side is', c)
end.

Perhaps it is not nice to have the types in the same font as variables. PASCAL can be extended with types, and therefore names of types are also variable.

2.9 How can we use the macros for typesetting program fragments in other languages?

Example *(Verbatims)*

The functionalities are verbatim text (in-line and in display), and file verbatim inclusion.

The markup starts with `\beginverbatim` and ends with `<escapechar>endverbatim`. The default escape character is '!'. 'Options,' for among other things numbering and enabling of metacode, are handled via the token variables `\thisverbatim`, or globally via `\everyverbatim`.

The place within context is controlled via the token variables `\preverbatim` and `\postverbatim`.

It is true that verbatims are heavily used by authors who write in (LA)TEX about (LA)TEX, and perhaps not so relevant for math authors.

[31] Courtesy Wilson and Addyman. Compare the markup with *The TEXbook*, 234 and ex18.9, and you will appreciate the approach of BLUe's format typesetting of PASCAL.

Example *(Verbatims: escape char use)*

```
%To handle via ! (escape char)
%   - other fonts
%   - emc (enable metacode)
\thisverbatim={\emc}
\beginverbatim
Some <metacode> and
blah, blah, ...    !it
Now text in italics!tt
and back again in tt
!endverbatim
```

```
Some ⟨metacode⟩ and
blah, blah, ...
Now text in italics
and back again in tt
```

Example *(Verbatims: line numbering and file inclusion)*

```
pretext
\everyverbatim={\numvrb}
\thisverbatim={%
   \catcode‘\!=12
   \catcode‘\|=12
%  \input vrb.tex %<file>
   \makeescape\*}%To terminate
\beginverbatim
Extras after file
*endverbatim
\noindent posttext
```

pretext

```
1 ⟨file⟩
2 Extras after file
```

posttext

Example *(Verbatims: line numbering)*

```
\thisverbatim={\numvrb
               \catcode‘\~=9
               \catcode‘\*=14 }
\beginverbatim
Just some text with
ligatures such as ¿ switched
off (TB 381).
line numbers restarted via
<escapechar>numvrb.!nonum
<escapechar>nonum switches off
line numbering.
<escapechar>vrblin<number>
assigns
<number> to the line counter.


!numvrb~After two blank*
lines.!nonum
!endverbatim
```

```
1 Just some text with
2 ligatures such as ¿ switched
3 off (TB 381).
4 line numbers restarted via
5 ⟨escapechar⟩numvrb.
  ⟨escapechar⟩nonum switches off
  line numbering.
  ⟨escapechar⟩vrblin⟨number⟩ assigns
  ⟨number⟩ to the line counter.


1 After two blanklines.
```

Example *(Verbatims: in line)*

```
\thisverbatim{\emc}
Before |<in-lineverbatim>| after.
```

Before ⟨*in-lineverbatim*⟩ after.

Also provided is `\unmc` to unable metacode.

For long verbatim listings supply

```
\everyverbatim{\baselineskip12ptplus2ptminus1pt\lineskip1ptplus1ptminus1pt}
```

to prevent underfull vbox messages.

2.10 How can we put `\it`, and `\tt` at the beginning of the lines?

2.11 How can we center verbatims?

2.12 A math display does not end a paragraph, it is part of the paragraph with the effect that paragraph parameters are not reset to their defaults. What about `\beginverbatim !endverbatim`?

Example *(Exercise and answer)*
Exercises and answers have to be marked up by

\exercise⟨*exercisetext*⟩
\answer⟨*answertext*⟩

%followed by a blank line

Example *(Footnotes)*
Plain TEX provides the `\footnote` command. *The TEXbook* ex15.12 deals with how to number a footnote automatically. In blue.tex the markup for the latter is `\ftn{`⟨*footnotetext*⟩`}`.
In the title part of an article—or report—BLUe falls back on `\footnote`, because those footnotes are some sort of out of order. That use reads as follows.

\footnote*{⟨*titlefootnotetext*⟩}
%or
\footnote{**}{⟨*titlefootnotetext*⟩}
%etc.

The above is regular TEX. Because of me using `\oldstyle` for the numbering the ⇀ symbol will be obtained instead of *. To correct this provide `{\rm *}`, or just forget about a marker and provide an empty group. From the context it is clear enough that the note is about the title. The footnotes are set in 8pt.

2.13 And what about notes in the margin?

Fonts and size switching

Default 10 pt is used. For the moment I stay with Knuth and copied his size-switching macros.[32] BLUe's format contains the size-switching macros `\Large`, ≈ 17pt, used for chapter headings, `\large`, ≈ 14.5pt, used for transparencies, and `\small`, that is, 8pt, used for footnotes. Next to these `\tenpoint`, `\ninepoint`, and `\eightpoint` have been inherited from manmac.
`\tiny` has size 5pt and has been setup similar to the `\...type` macros from Knuth's concert format, Appendix E of *The TEXbook*. A pedagogical stepping stone for size-switching macros, and sufficient for just text.

```
\def\tiny{\fiverm\baselineskip7pt\let\it\fivei\let\bf\fivebf\let\oldstyle\it}
```

2.14 How to setup a general `\twelvepoint` size-switching macro?

2.15 What is the effect of `\bf\it`? And of `\it\bf`?

2.16 Experiment with the difference between `\twelvepoint` and the use of `\magstep1`.[33]

[32] Because of this attitude I could easily kameleon `\oldstyle` to all those sizes. To achieve this with NFSS would have cost me much more time, I guess.
[33] Courtesy Nico Temme.

Mathematics

On top of plain TEX blue.tex provides facilities for: automatic numbering, cross-referencing for formulas, matrix icons, multi-alignment points in eqalign, primed summation symbols, arrows for commutative diagrams, and some special symbols. Examples borrowed from literature are included to illustrate how TEX can be used to mark up mathematics. This anthology shows that many details have to be prescribed. To give a math manuscript to a keyboarder without markup guidance is doomed to yield mediocre results.

Swanson 1986 is a good source for what math should look like in print. Very nice is also 'Mathematical Writing,' a report from a Stanford Workshop organized by Knuth. TEX is already very rich for math markup, because in his Preface to *The TEXbook* Knuth states

> '... TEX, a new typesetting system intended for the creation of beautiful books—and especially for books that contain a lot of mathematics.'

What's the problem, Doc?

If we assume that mathematicians write math manuscripts in the classical sense, then there is a problem when those mathematicians wish to have their work formatted via computer-assisted document-preparation tools like TEX. What has to be keyboarded looks much different from the manuscript. Examples of this phenomenon are omni-present in this work.

Example *(Disparity math notation and markup notation)*

```
x=1+\left({y^2\over k+1}
      \right)^{\!\!1/3}
```

$$x = 1 + \left(\frac{y^2}{k+1}\right)^{\!\!1/3}$$

Because of this disparity, the problem is how to get a correct[34] TEXscript, starting from a mathscript. This is difficult[35] due to the inherent complexity of math typesetting, and due to the unusual nature of TEX, if not because of the bewildering and confusing flavours of TEX-based products. Even a fancy and friendly, WYSIWYG user-interface is not enough. Optical scanners of math—or systems which understand spoken mathematics—are still science fiction.

The extras

blue.tex provides facilities for: automatic numbering, cross-referencing for (display) formulas, macros for matrix icons, an extension of `\eqalign` with respect to multi-alignment points, primed summation symbols, arrows for commutative diagrams, `\beginmathdemo` and `\endmathdemo` from manmac, and poor man's blackboard bold and some other special symbols. These extras are introduced via examples.

---

34 Not only in the sense that the TEX formatter does not complain, no, correct in the sense of complying with tradition of mathematical typesetting.

35 In the example at hand the keyboarder has to be aware of `\over` and / respectively, and when to use which. Moreover, the correct size of the parentheses must be supplied, and some kernings have to be inserted!

Example *(Automatic numbering and cross-referencing)*

This is a compatible extension. For the markup of math cross-referencing insert instead of plain's explicit (reference) number

- `\ref`, for creation of the number, and `\ref\`⟨*name*⟩, for attaching a name to the automatically generated number, and
- `\crsref\`⟨*name*⟩, for cross-referencing.

```
%Automatic numbering
$$a+b\eqno\ref$$
```

$$a+b \qquad (1)$$

```
%Automatic numbering, and
%symbolic cross-referencing
$$c*d\eqno\ref\cgl$$
Text, \crsref\cgl
```

$$c*d \qquad (2)$$

Text, (2)

3.1 And what about forward referencing?

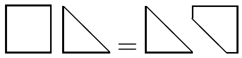
Handy token variables are `\prenum` and `\postnum`.[36] Their contents is inserted before, respectively after, the automatically generated number in `\ref` and `\crsref`. Because of this one can get labels and crossrefs like 13a, or enclosing numbers by parenthesis. The latter are the defaults of these token variables. If in such a case the formula counter must keep its value, provide `\advancefalse`.

3.2 How can we retain the number, and suffix a letter, as label of a formula to come?

Example *(Matrix icons)*

Useful icons concern the matrices: rectangular, via `\icmat`, triangular (lower left and upper right), via `\iclt` or `\icurt`, and upper Hessenberg, via `\icuh`. The arguments are dimensionless numbers. The first argument reflects the vertical size, and the second the horizontal size. In case of `\icuh` the second argument is the Hessenberg bandwidth and the third is the difference between the first and the second.

```
\unitlength1ex
$$\icmat44\kern\unitlength\iclt44=
    \iclt44\icuh413\qquad
    \hbox{AL}=\hbox{LH}$$
```



AL = LH

3.3 Another matrix factorization reads

```
$$\icmat63=\icmat63\kern\unitlength
  \icurt63\qquad\hbox{A}=\hbox{QR}$$
```

When used together with the example above, align on the =-signs, that is, on the =-sign in the icons and on the =-sign in the formulas.

Example *(Compatible extension of eqalign with multiple alignment points)*

```
$$\eqalign{
\cos(z\sin\theta)={}&
  J_0(z)&
  {}+2\sum_{n=1}^\infty
    J_{2n}(z)\cos2n\theta\cr
  \sin(z\sin\theta)={}&
        &
  {}+2\sum_{n=1}^\infty
J_{2n+1}(z)\sin(2n+1)\theta\cr
}$$
```

$$\begin{aligned}\cos(z\sin\theta) &= J_0(z) + 2\sum_{n=1}^\infty J_{2n}(z)\cos 2n\theta\\ \sin(z\sin\theta) &= \qquad\quad\; + 2\sum_{n=1}^\infty J_{2n+1}(z)\sin(2n+1)\theta\end{aligned}$$

[36] Courtesy Michael Spivak.

3.4 Why is the empty formula used in the markup?

Example *(Macros for showing math markup and the result)*

`\beginmathdemo` (and variants) and `\endmathdemo`, are used in the *The TeXbook* script, 444–466, for indented display Math, see *The TeXbook* chapters 16–19.[37] They are used to typeset the marked up copy and the typeset result side-by-side. The user does not have to bother about the template for the alignment display used. The functionality provided is similar, but a little restricted, to the (LaTeX) styles for switching from one-column to two-column format and vice versa. But, ... it is much simpler and more efficient—IMHO with all respect—because it does not entail OTR processing.

```
%TeXbook 128
\beginmathdemo
   \it Input&\it Output\cr
   \noalign{\vskip2pt}
   |$x^2$|&x^2\cr
\endmathdemo
```

Input	*Output*
`$x^2$`	x^2

```
%TeXbook 139
\begindisplaymathdemo
   |$$x+y^2\over k+1$$|&
           x+y^2\over k+1\cr
   \noalign{\vskip2pt}
   |$${x+y^2\over k}+1$$|&
           {x+y^2\over k}+1\cr
\endmathdemo
```

`$$x+y^2\over k+1$$`	$\dfrac{x+y^2}{k+1}$
`$${x+y^2\over k}+1$$`	$\dfrac{x+y^2}{k}+1$

Example *(Poor man's blackboard bold and some special symbols)*

Now and then other symbols than those provided in the font tables of Appendix F of *The TeXbook* are needed.[38] These can be constructed approximately.

```
$$\halign{#\hfil\quad&
              \hfil#\hfil\cr
 natural numbers    &$\IN$ \cr
 integers           &$\Z$ \cr
 rational numbers   &$\Q$ \cr
 reel numbers       &$\R$ \cr
 and complex numbers&$\C$ \cr
 next to\cr
 greater or less &$\gtrless$\cr
 external tensor product&
               $\boxtimes$\cr}$$
```

natural numbers	$\mathbb{N}$
integers	$\mathbb{Z}$
rational numbers	$\mathbb{Q}$
reel numbers	$\mathbb{R}$
and complex numbers	$\mathbb{C}$
next to	
greater or less	$\gtrless$
external tensor product	$\boxtimes$

`\IN`, `\Z`, et cetera are incorporated in blue.tex.

3.5 On the TeX-NL network there was a request for the pro mille token. How to get it?

Use

Many ingredients are supplied by plain. It is just a matter of what-and-how, what to use from the wealth offered. It has all to do with what the script should look like in print *within the context*. In the sequel some more math markup will be shown, not restricted to markup tags from blue.tex. A classical example is the markup for the various uses of O.

Noteworthy is further that punctuation symbols are also used with spacing before them, ditto for the vertical bars and the backslash. The size of delimiters is dependent on the

[37] There is only one second part macro for all these cases. Its replacement text is modified into `\crcr\egroup$$`.

[38] Generally, non-standard fonts are already available somewhere. For math consult AMS.

context, which can't be completely automated. The variants can be obtained via special markup, for example via

- coercion macros (to guide TEX with respect to spacing before and after, for example via `\mathdelimiter`, or to positioning of embellishments below and on top, for example via `\mathop`)
- the use of the empty formula, {}, to coerce binary behaviour of the operator
- macros to impose the size of delimiters (`\biggl` et cetera)
- special control sequences (like `\colon`).

Plain's display maths Most displays belong to one of the categories given below.

Example *(A labeled formula in display)*

Spaces are neglected in math mode. The kern `\,` is needed to coerce the correct spacing.

```
$$\sin2x=2\sin x\,\cos x
     \eqno({\rm TB186})$$
```

$$\sin 2x = 2\sin x\,\cos x \qquad \text{(TB186)}$$

Example *(Formula hyphenation; shifting of lines)*

A hyphenated formula via `\displaylines`. With the use of `\hfill` we can shift lines to the left or right.

```
$$\displaylines{F(z)=
a_0+{a_1\over z}+{a_2\over z^2}
  +\cdots+{a_{n-1}\over z^{n-1}}
  +R_n(z),\hfill\cr
\hfill n=0,1,2,\dots\,,\cr
F(z)\sim\sum_{n=0}^\infty
  a_nz^{-n},\quad z\to\infty
  \hfill\llap{(TB ex19.16)}}$$
```

$$F(z) = a_0 + \frac{a_1}{z} + \frac{a_2}{z^2} + \cdots + \frac{a_{n-1}}{z^{n-1}} + R_n(z),$$
$$n = 0, 1, 2, \ldots\,,$$
$$F(z) \sim \sum_{n=0}^{\infty} a_n z^{-n}, \quad z \to \infty \qquad \text{(TB ex19.16)}$$

Example *(Alignment and centered labeling)*

```
$$\eqalign{\cos2x
      &=2\cos^2x-1\cr
      &=\cos^2x-\sin^2x}
  \eqno({\rm TB193})$$
```

$$\begin{aligned}\cos 2x &= 2\cos^2 x - 1\\ &= \cos^2 x - \sin^2 x\end{aligned} \qquad \text{(TB193)}$$

Example *(Alignment and labels per line)*

```
$$\eqalignno{\cosh2x
     &=2\cosh^2x-1&({\rm TB192})\cr
     &=\cosh^2x+\sinh^2x}$$
```

$$\begin{aligned}\cosh 2x &= 2\cosh^2 x - 1 && \text{(TB192)}\\ &= \cosh^2 x + \sinh^2 x\end{aligned}$$

Remark. If one wants `\eqalignno` to behave like `\displaylines`—that is, left-justified—then modify in `\eqalignno` the first `\tabskip=\centering` assignation into `\tabskip=\z@skip`.

Example *(Subscripts)*

The depth of a subscript depends on whether there is a superscript. With a superscript a subscript sinks a little. In order to obtain uniformly placed subscripts the solution is to adjust the following font dimension parameters, see *The TEXbook* 179.

```
\fontdimen16\textfont2=2.7pt
\fontdimen17\textfont2=2.7pt
$$X_1+Y_1^2=1$$
```

$$X_1 + Y_1^2 = 1$$

A snapshot of examples borrowed from *The TEXbook*, to illustrate the need for extra markup. *The TEXbook* chapters 16–19 contain many examples, well-ordered and appropriately explained.

Example *(Dots and the comma after)*

```
$${\bf S^{\rm-1}TS=dg}(\lambda_1,
   \ldots\,,\lambda_n)=\bf\Lambda$$
```

$${\bf S^{\rm-1}TS=dg}(\lambda_1,\ldots\,,\lambda_n)=\bf\Lambda$$

Example *(Summation with limits)*

```
$$\sum_{k=1}^\infty{1\over2^k}=1$$
```

$$\sum_{k=1}^\infty{1\over2^k}=1$$

3.6 In line we usually have subscripts and superscripts. How can we get those?

Example *(Overlining)*

If there is an example with various markup possibilities, this is the one, although the various O-s comes close.

```
$$\bar z,\ \overline z,\
\bar P,\ \overline P,\
\bar h,\ \hbar,\
\overline{AB}$$
```

$$\bar z,\ \overline z,\ \bar P,\ \overline P,\ \bar h,\ \hbar,\ \overline{AB}$$

Example *(Square roots)*

```
$$\sqrt{1+\sqrt{1+\sqrt{1+x}}}$$
```

$$\sqrt{1+\sqrt{1+\sqrt{1+x}}}$$

Example *(Roman texts in math)*

```
$${f(x+\Delta x)-f(x)\over
     \Delta x}\to f'(x)\quad
  {\rm as}\quad\Delta x\to0$$
```

$${f(x+\Delta x)-f(x)\over \Delta x}\to f'(x)\quad {\rm as}\quad\Delta x\to0$$

Example *(Kerning; positive and negative)*

```
$$         \int\!f(x)\,dx,   \quad
 \Gamma_{\!2}+\Delta^{\!2},\quad
 \sum^\infty_{n=-\infty}\!
                 \!\!\cos nt$$
```

$$\int\!f(x)\,dx,\quad \Gamma_{\!2}+\Delta^{\!2},\quad \sum^\infty_{n=-\infty}\!\!\!\cos nt$$

Example *(Empty formula and subscripting)*

```
(\lambda)_2\,{}_2F_1
```

$(\lambda)_2\,{}_2F_1$

Example *(Math operator)*

```
$$\mathop{\hbox{\rm Res}}_
  {s=e^{i\pi}}f(s)=-e^{i\pi z}$$
```

$$\mathop{\rm Res}_{s=e^{i\pi}}f(s)=-e^{i\pi z}$$

Example *(Colon markup; punctuation vs. operator)*

```
$$f\colon A\to B,\quad \{x:x>5\}$$
```

$$f\colon A\to B,\quad \{x:x>5\}$$

Example *(Context-dependent size)*

```
$$\bigl\|\,\alpha(\sqrt
   {\mathstrut a}+\sqrt
   {\mathstrut b}\,)\,
  \bigr\|
\leq|\alpha|\,
  \bigl\|\sqrt
   {\mathstrut a}+\sqrt
   {\mathstrut b}\,
  \bigr\|$$
```

$$\bigl\|\,\alpha(\sqrt{\mathstrut a}+\sqrt{\mathstrut b}\,)\,\bigr\|\leq|\alpha|\,\bigl\|\sqrt{\mathstrut a}+\sqrt{\mathstrut b}\,\bigr\|$$

It is tempting to insert a multiplication dot. Don't!

Example *(Vertical bars, The TeXbook 146, 147, ex18.21)*

```
$$\big\{\,x^3\bigm\vert h(x)\in
       \{-1,0,+1\}\,\bigr\}$$
```

$$\big\{\,x^3\bigm\vert h(x)\in\{-1,0,+1\}\,\bigr\}$$

Example *(Halves variety The TeXbook ex11.6)*

```
$$D^\lambda_0(z)=
  4a_\lambda\, z\,{}_2F_1(%
  \textstyle
  \lambda+{1\over2},{1\over2};
  {3\over2};z)$$
%Typographer's 1/2
Typographer's $\fracdek1/2$,
(recipes), which works better
than a mathematician's $1\over2$
```

$$D^\lambda_0(z)=4a_\lambda\, z\,{}_2F_1(\textstyle\lambda+{1\over2},{1\over2};{3\over2};z)$$

Typographer's ${}^1\!/_2$, (recipes), which works better than a mathematician's $\frac12$

Example *(Under and overbraces)*

Subtle use of fonts, and `\mathstrut` to enforce size. For under and over parentheses see TTN 3.4.

```
$$\{\underbrace{\overbrace
 {\mathstrut a,\ldots,a}^
     {k\;a\mathchar'27\rm s},
 \overbrace{\mathstrut b,\ldots
    ,b}^{l\;b\mathchar'27\rm s}}
      _{k+l\rm\;elements}\}$$
```

$$\{\underbrace{\overbrace{\mathstrut a,\ldots,a}^{k\;a\text{'s}},\overbrace{\mathstrut b,\ldots,b}^{l\;b\text{'s}}}_{k+l\rm\;elements}\}$$

Example *(Diagonal dots, coercions, The TeXbook ex18.45)*

```
$$2\uparrow\uparrow k
   \mathrel{\mathop=^{\rm def}}
   2^{2^{2^{\cdot^{\cdot^
            {\cdot^2}}}}}\vbox
  {\hbox{$\Big\}\scriptstyle k$}
   \kern0pt}$$
```

$$2\uparrow\uparrow k\mathrel{\mathop=^{\rm def}}2^{2^{2^{\cdot^{\cdot^{\cdot^2}}}}}\Big\}{\scriptstyle k}$$

3.7 What is the function of the `\kern0pt`?

3.8 Can the `\cdot` be replaced by just a period?

Example *(Undoing mathsurround space)*

```
$2{\times}3$-matrix
```

$2{\times}3$-matrix

Example *(All those Os)*

```
$\emptyset$, (the empty set)
$f\circ g\colon
  x\mapsto f\bigl(g(x)\bigr)$,
  (composition), and
   order symbols
$o(h^2)$,
$O(h^2)$.
```

$\emptyset$, (the empty set),
$f\circ g\colon x\mapsto f\bigl(g(x)\bigr)$ (composition),
and the order symbols $o(h^2)$ and $O(h^2)$

Example *(Set difference vs. cosets)*

```
$A\setminus A=\emptyset,
 \hbox{and the cosets of $G$
 by $H$:\ }G\backslash H$
```

$A\setminus A=\emptyset,\hbox{and the cosets of G by H:\ }G\backslash H$

Example *(The Cardano solution to third-order equation)*

```
%x^3+px=q, p,q\ge0
$$\root3\of{\sqrt{p^3/27-q^2/4}+
  q/2}-\root3\of{\sqrt{p^3/27+
  q^2/4}-q/2}$$
```

$$\sqrt[3]{\sqrt{p^3/27-q^2/4}+q/2}-\sqrt[3]{\sqrt{p^3/27+q^2/4}-q/2}$$

Example *(Derivatives)*

The problem is the three dotted derivative.

```
$$\dot y\,\ddot y\,
  \dot{\ddot y\kern2pt}\quad
  y'\,y''\,y'''         \quad
\partial_xy\,\partial_x^2y\,
\partial_x^3y$$
```

$$\dot y\,\ddot y\,\dddot y\quad y'\,y''\,y'''\quad \partial_xy\,\partial_x^2y\,\partial_x^3y$$

Example *(Bessel equation)*

```
$$z^2w''+zw'+(z^2-\nu^2)w=0$$
solutions:
$J_{\pm\nu}(z)$,
$Y_{\pm\nu}(z)$,
$H_\nu^{(1)}(z)$,
$H_\nu^{(2)}(z)$
```

$$z^2w''+zw'+(z^2-\nu^2)w=0$$

solutions: $J_{\pm\nu}(z)$, $Y_{\pm\nu}(z)$, $H_\nu^{(1)}(z)$, $H_\nu^{(2)}(z)$

Example *(Primed summation symbols and split formula)*

Subtle use of the prime and the right font. Nice is the nested use of the coercions `\mathop` and `\mathrel`. A minor detail is to preserve the dyadic character of the + in the last term. `\acclap` is incorporated in blue.tex. See *The TEXbook* ex18.44.

```
$$\displaylines{
  \mathop{{\sum}\acclap'}_{k=0}^n
  a_kT_k(x)
  \mathrel{\mathop=^{\rm def}}
 .5\kern1pt a_0+a_1 x+a_2T_2(x)+
  \cdots                  \hss\cr
  \hfill{}+a_nT_n(x)}$$
```

$$\sum_{k=0}^{n}{}' a_kT_k(x) \stackrel{\rm def}{=} .5\,a_0+a_1x+a_2T_2(x)+\cdots$$
$$+\,a_nT_n(x)$$

3.9 How can we prevent the line distance from growing larger than the regular value?

Example *(Hypergeometric function)*

Subtle subscripting, size of parentheses, and positioning of arguments.

```
$$M_n(z)={}_{n+1}F_n\Bigl({k+a_0,
   \atop\phantom{kc_1}}
   {k+a_1,\dots,k+a_n
   \atop k+c_1,\dots,k+c_n};z\Bigr)
$$
```

$$M_n(z)={}_{n+1}F_n\Bigl({k+a_0,\atop}{k+a_1,\dots,k+a_n\atop k+c_1,\dots,k+c_n};z\Bigr)$$

3.10 Why has the `\phantom` argument `kc_1`?

Example *(From The TEXbook 177, (p)matrix as formula part. proclaim is used too)*

```
\proclaim Definition. $x$ is called
an eigenvector with
eigenvalue $\lambda$ of the matrix
$$A=\pmatrix{
    a_{11}&a_{12}&\ldots&a_{1n}\cr
    a_{21}&a_{22}&\ldots&a_{2n}\cr
    \vdots&\vdots&\ddots&\vdots\cr
    a_{n1}&a_{n2}&\ldots&a_{nn}\cr}$$
if $Ax=\lambda x$.
\par
```

Definition. *x is called an eigenvector with eigenvalue λ of the matrix*

$$A=\begin{pmatrix} a_{11}&a_{12}&\ldots&a_{1n}\\ a_{21}&a_{22}&\ldots&a_{2n}\\ \vdots&\vdots&\ddots&\vdots\\ a_{n1}&a_{n2}&\ldots&a_{nn}\end{pmatrix}$$

if $Ax=\lambda x$.

Example *(Split equation and context-sized delimiters)*

```
From Swanson (1986, Section 3.3
              Math in Display)
Because it is a large formula
I used \displaylines.
For the integral we need \nolimits
to inactivate the default placement
of limits.
Furthermore, there is subtle use of
subscripting and delimiters of
varying sizes.
Finally the shifting of parts
has to handled corrrectly.
See the script for the details.
```

$$\begin{gathered}\int_U \delta(I)\mu(I) \le \\ \sum_{\mathcal{D}}\sum_{\mathcal{D}_{I'}}\biggl[\int_J \alpha(J')\mu(J') - \alpha(J)\mu(J) \\ \hfill - \int_J [\{s(\alpha\eta)(J')\}/\eta(J')]\mu(J')\biggr] \\ + \biggl[\sum_{\mathcal{D}}\sum_{\mathcal{D}_{I'}}\alpha(\mathrm{J}) - [\{\mathtt{s}(\alpha\eta)(\mathrm{J})\}/\eta(\mathrm{J})]\mu(J)\biggr] \\ \times\biggl[\sum_{\mathcal{D}}\sum_{\mathcal{D}_{I'}}\alpha(\mathrm{J}) - [\{\mathtt{s}(\alpha\eta)(\mathrm{J})\}/\eta(\mathrm{J})]\eta(J)\biggr]\end{gathered}$$

Example *(Rhombus scheme)*

```
The idea is to align vertically.
Pseudo markup reads
\setbox\ru={⟨/ line⟩}
\setbox\rl={⟨\ line⟩}
$$\halign{⟨template⟩\cr
1st &   &e...\cr
    &\ru&    &\rl&    \cr
q...&   &    &   &q...\cr
    &\rl&    &\ru&    \cr
    &   &e...&\omit...
              \hidewidth\cr}$$
For the details consult the script.
```

$$\begin{array}{ccccc} 1^{st}\mathrm{RS} & & e_k^{(s)} & & \\ & / & & \backslash & \\ q_k^{(s+1)} & & & & q_{k+1}^{(s)} \\ & \backslash & & / & \\ & & e_k^{(s+1)} = \frac{q_{k+1}^{(s)}}{q_k^{(s+1)}}\, e_k^{(s)} & & \\ 2^{nd}\mathrm{RS} & & q_k^{(s)} & & \\ & / & & \backslash & \\ e_{k+1}^{(s+1)} & & & & e_k^{(s)} \\ & \backslash & & / & \\ & & q_k^{(s+1)} = q_k^{(s)} + (e_k^{(s)} - e_{k+1}^{(s+1)}) & & \end{array}$$

Example *(Commutative diagram, confer The TEXbook ex18.46)*

Nice use of two-sided `\hidewidth`, and the subtle `\strut` to place the superscript of $\mathcal{F}$. There is no strict alignment. The elements at the nodes overflow into the space of the arrows. As a consequence we need arrows of various lengths.

```
\let\normalbaselines
    \adaptedbaselines
$$\matrix{
  f&\lmapright\otimes&a_f\cr
  \mapdown{{\cal F}}&&\mapup{%
           {\cal F}\strut^{-}}\cr
  \hidewidth{\cal F}(f)\hidewidth
    &\mapright\times\hfil&
    \hidewidth\bigl({\cal F}(f)
         \bigr)^2\hidewidth\cr}$$
```

$$\begin{array}{ccc} f & \xrightarrow{\otimes} & a_f \\ \Big\downarrow{\mathcal F} & & \Big\uparrow{\mathcal F^-} \\ \mathcal{F}(f) & \xrightarrow{\times} & \bigl(\mathcal{F}(f)\bigr)^2 \end{array}$$

Example *(Partitioning)*

```
$$\def\data{$I_{n-r}$\cs 0       \rs
                0\cs $I-2v_rv_r^T$}
P_r=\left(\vcenter{\ruled
          \btable\data}\right)$$
```

$$P_r = \left(\begin{array}{c|c} I_{n-r} & 0 \\ \hline 0 & I - 2v_r v_r^T \end{array}\right)$$

3.11 How can we avoid \$-s in the markup of the data?

3.12 How can we construct partitioned matrices in general?

Example *(Continued fractions with alignment on =, and interruption)*

```
Many details come together in
this markup. Not in the least
the alignment on = where one is
an =-def. Also the space
saving variants, borrowed from
literature are relevant.
The special ‘\over’-line is a
poor man’s use of \atop and
\over, because \over is a
primitive.
Peruse the script.
```

$$1 + \mathop{\Phi}_{k=1}^{n} \frac{a_k}{b_k} \stackrel{\rm def}{=} 1 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \ddots + \cfrac{a_{n-1}}{b_{n-1} + \cfrac{a_n}{b_n}}}}$$

with (space saving) variant notations

$$\begin{aligned} &= 1 + \frac{a_1|}{|b_1} + \frac{a_2|}{|b_2} + \cdots + \frac{a_n|}{|b_n} \\ &= 1 + \frac{a_1}{b_1 +}\,\frac{a_2}{b_2 +} \cdots \frac{a_n}{b_n} \end{aligned}$$

3.13 How can we replace the division line by ┌┘?

Example *(Matrix equation)*

```
$$\displaylines{\indent
\bordermatrix{
   &      &\rm A &      \cr
   &\times&\times&\times\cr
   &\times&\times&\times\cr
   &\times&\times&\times}
\bordermatrix{
   & &\rm N & \cr
   &1&      & \cr
   &0&1     & \cr
   &0&\times&1}
\hfill\cr\hfill=
\bordermatrix{& &\rm N & \cr
              &1&      & \cr
              &0&1     & \cr
              &0&\times&1}
\bordermatrix{
   &      &\rm H &      \cr
   &\times&\times&\times\cr
   &\times&\times&\times\cr
   &0     &\times&\times}
}$$
```

$$\begin{array}{c} \mathrm{A} \\ \begin{pmatrix}\times&\times&\times\\ \times&\times&\times\\ \times&\times&\times\end{pmatrix}\end{array} \begin{array}{c} \mathrm{N} \\ \begin{pmatrix}1&&\\ 0&1&\\ 0&\times&1\end{pmatrix}\end{array}$$

$$= \begin{array}{c} \mathrm{N} \\ \begin{pmatrix}1&&\\ 0&1&\\ 0&\times&1\end{pmatrix}\end{array} \begin{array}{c} \mathrm{H} \\ \begin{pmatrix}\times&\times&\times\\ \times&\times&\times\\ 0&\times&\times\end{pmatrix}\end{array}$$

Example *(Braces and matrices)*

```
This is a complex problem, with a lot
of fine-tuning. However, it is
possible to systemize much.
The approach is to look at it as
a bordered table with a
parenthesized matrix as data.
The handling of the row stub and
the header are separated from
the data (the pmatrix).
The pseudo code reads as follows.
\setbox1=\hbox{$\pmatrix{⟨data⟩}$}
Define header and row stub list, and
switch off defaults of btable,
especially the separators.
Finally invoke
$$\btable{$\vcenter{\copy1}$}$$.
For the details see the script.
```

$$\begin{array}{rc} & \overbrace{\hphantom{\times\quad\times\quad\times\quad\times}}^{p}\quad\overbrace{\hphantom{\times\quad\times\quad\times}}^{n-p} \\ \begin{array}{r} p\left\{\vphantom{\begin{array}{c}0\\0\\0\\0\end{array}}\right. \\ n-p\left\{\vphantom{\begin{array}{c}0\\0\\0\end{array}}\right. \end{array} & \begin{pmatrix} \times&\times&\times&\times&\times&\times&\times\\ 0&\times&\times&\times&\times&\times&\times\\ 0&0&\times&\times&\times&\times&\times\\ 0&0&0&\times&\times&\times&\times\\ 0&0&0&0&\times&\times&\times\\ 0&0&0&0&\times&\times&\times\\ 0&0&0&0&\times&\times&\times \end{pmatrix} \end{array}$$

3.14 Add partitioning to the braces and matrices example.

Tabular Material

BLUe's format provides btable for the class of (simple) bordered tables. It will be shown that this model can be used to format a wide variety of tables. Difference in representation can be obtained via attributes, i.e., without changing the markup. Customization of the macro can be done by modification of the separators for the caption, the columns, and the like. The diversity of tabular material in general has been illustrated by a kaleidoscope of tables borrowed from literature. For the latter group the markup is not necessarily restricted to btable or halign. The example of how Knuth marked up real-life tables is illustrated by his font charts.

The backbone of blue.tex's tables is `\btable`, to format a bordered table. Other TEXniques for formatting tables will be dealt with too. The formatting of tables with simultaneous row and column spans is a subject for the 'educated layman,' not for the faint-hearted.

Bordered table

A bordered table consists of a caption, a header, row stubs, a table proper, and a footer, all with parameterized separators. Special cases occur when some of the elements are absent.

Abstraction from a ruled and a non-ruled table is realized at the user level. Ruled can be specified either as horizontal, vertical or both. Dotted rules, also the partial ones, can be specified. No variation in the line thicknesss has been coded. Framing has been treated just as if a general document element has to be framed. I chose to leave the caption and the footer outside the frame.[39] The positioning of the elements can be specified as centered, flushed left, or flushed right. A template is provided by default. Caption, border (first, header, row stubs), and footer can be specified; all mutual orthogonal and independent of `\data`.

Model

← C a p t i o n →

First	Header
r o w s t b l s t	Table Proper

← F o o t e r →

Markup tags

- tags `\beginbtable` and `\endbtable`, or the short-cut `\btable`, and the column and row separator `\cs`, and `\rs`
- defs to be redefined `\btablecaption`, `\first`, `\header`, `\footer`, `\rowstblst`, (all defined empty by default)

---

[39] However, we can frame any document element, can't we?

- table markup tags, `\logms`, logical multispan to span a number of *logical* columns
- attributes
  - `\ruled`, `\hruled`, `\vruled`, `\dotruled` Default: `\nonruled`
  - `\framed`, `\dotframed` Default: `\nonframed`
  - `\fll` (flushed left), `\flr` (flushed right) Default: `\ctr`
- token variables `\prebtable`, `\postbtable` for positioning the table within the context.[40]

By the way the above model is an example of the markup of a bordered table.[41] The markup reads as follows.

```
\def\btablecaption{$\leftarrow$\hfill C a p t i o n\hfill$\rightarrow$}
\def\first{\hbox to6ex{\hss First\hss}}
\def\header{\hbox to18ex{\hss Header\hss}}
\def\rowstblst{{\hfil$\vcenter to20ex{\vss\offinterlineskip
                \fifo rowstblst\ofif\vss}$}}
\def\process#1{\hbox to0pt{\hss#1\hss}\kern.5ex}
\def\data{$\vcenter{\offinterlineskip\hbox{Table}\kern1ex\hbox{Proper}}$}
\def\footer{$\leftarrow$\hfill{\small F o o t e r}\hfill\hfill\hfill
            $\rightarrow$}}
$$\vcenter{\btable\data}$$
```

Customization \⟨*X*⟩`sep` are parameters, which govern the kind of separators. The following are defaults to be customized.

```
\def\cellsize{\cvsize=4ex\tsht=.775\cvsize\tsdp=.225\cvsize}
\def\colsepsurround{\kern.5em}\def\captionsep{\medskip}\def\rowstbsep{\vrule}
\def\headersep{\hrule}\def\footersep{\smallskip}\def\template{}
```

The positioning of the elements in the cells is parameterized by `\lft` and `\rgt`. They are equal to `\hfil` by default. A nice application is to make these equal to $ in such a way that data can be processed in math mode.

The empty template looks curious. The explanation is: when empty—i.e., a user did not provide one—a standard template will be used.

Use of btable macro

There are some pitfalls that I know of and that a user might like to be aware of.

- When `\header`, `\footer` and the like are defined they keep their value. So when you don't have a header in the next table, be sure that this definition is empty.
- Most of the time I used definitions with `\btable` instead of token variables. Beware.
- To specify the row stub list separately from the data might yield a first column and a data part of unequal size.
- The `\cs` is ambiguous. In blue.tex it has the function to yield the argument as control sequence, while within btable the meaning is the column separator.
- When verbatims are also part of the data process the table on the fly via `\beginbtable` ... `\endbtable`. However, there is a problem when the header and footer have to be processed on the fly too.
- In the markup of the table in the appendix 'History of Changes,' the problems are the right line distances in a `\vtop`. The latter as element in the table. It is caused by the `\tstrut`, table strut, which is inserted at the end of each table row. See the script how to circumvent this problem.
- The attribute `\dotruled` is not so general as `\ruled` or `\vruled`, because we lack the primitive `\vdotfill` as a dotted alternative for `\vrule`. For horizontal rules TEX provides the primitives `\hrule` with the dotted variant `\dotfill`.

---

[40] In practice the math display is used most of the time.

[41] Admittedly a little complicated. Simpler examples will follow.

In the examples of use below the invocation of `\data` could have been hidden. However, in order to remain compatible with the situation where a user wishes to forget about `\data` and provide the table entries directly as argument to `\btable`, I refrained from that.

Example *(Just data, magic squares)*

2	7	6
9	5	1
4	3	8

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

via

```
$$\def\data{2\cs7\cs6\rs
            9\cs5\cs1\rs
            4\cs3\cs8}
\vcenter{\framed\ruled\btable\data}\qquad\qquad\qquad
  \def\data{16\cs    3\cs    2\cs13\rs
             5\cs   10\cs   11\cs 8\rs
             9\cs    6\cs    7\cs12\rs
             4\cs   15\cs   14\cs 1}
 \def\template{\colsepsurround\lft{\oldstyle##}\rgt
               &&\lft##\rgt&\lft{\oldstyle##}\rgt\cr}
 \ruled\framed\setbox0=\btable\data%for measuring sizes
 \vcenter{\dotruled\dotframed\btable\data}\qquad$$
```

Remark. To introduce the `\oldstyle` elegantly I added a template for the `\halign`. For doing this, awareness of the physical columns is needed. Logically we are talking about a 3∗3 and 4∗4 table. Note that for dotted variants much more work has to be done.

Example *(Data plus header, The TEXbook 246)*

Year	World Population
8000 B.C.	5,000,000
50 A.D.	200,000,000

Year	World Population
8000 B.C.	5,000,000
50 A.D.	200,000,000

via the nearly descriptive markup

```
\def\header{\hfill Year\hfill\cs \hfill World Population\hfill}
\def\data{8000 B.C.\cs  5,000,000\rs 50 A.D.\cs200,000,000}
$$\flr\vruled\vcenter{\btable\data}\qquad\ruled\framed\vcenter{\btable\data}$$
```

4.1 How come that the left table contains a `\hrule`?

Example *(Data with row stubs, The TEXbook 232)*

```
\def\rowstblst{{Horizontal lists}
     {Vertical lists}{Math lists}}
\def\rowstbsep{}%Customization
\def\data{Chapter 14\rs
          Chapter 15\rs
          Chapter 17}
$$\btable\data$$
```

Horizontal lists	Chapter 14
Vertical lists	Chapter 15
Math lists	Chapter 17

4.2 How can we obtain the same table without the use of `\rowstblst`?

Example *(Data, row stubs, header, caption and footer)*

11	12
21	22

	Header	
1^{st} row	11	12
2^{nd} row	21	22

Caption

	Header	
1^{st} row	11	12
2^{nd} row	21	22

Footer

Caption

	Header	
1^{st} row	11	12
2^{nd} row	21	22

Footer

after data definition, `\def\data {11\cs12\rs 21\cs22}`, via[42]

```
$$\vcenter{\framed\btable\data}        \qquad
     \def\header{\logms2\hfill Header\hfill}
     \def\rowstblst{{$1^{st}$ row}
                    {$2^{nd}$ row}}
  \vcenter{\btable\data}               \qquad
    \def\btablecaption{Caption}
    \def\footer{Footer}
  \vcenter{\dotruled\btable\data}      \qquad
  \vcenter{\ruled\framed\btable\data}$$
```

The above example shows in a nutshell the essential possibilities of the `\btable` macro. It is my favorite, although other nice and powerful examples are still to come.

Example *(Walter's spreadsheet, The TEXbook 244)*

No preamble has to be defined.

1	Adjusted gross income		\$4,000
2	Zero bracket amount for		
	a single individual	\$2,300	
3	Earned income	1,500	
4	Substract line 3 from line 2		800
5	Add lines 1 and 4. Enter here		
	and on Form 1040, line 35		\$4,800

The above is marked up, as a 4-column table, via

```
\def\data{1\cs\logms2Adjusted gross income     \dotfill\cs \$4,000\rs
          2\cs Zero bracket amount for                          \rs
           \cs a single individual \dotfill\cs\hfill\$2,300\cs    \rs
          3\cs Earned income       \dotfill\cs
                                    \hfill\underbar{ 1,500}\cs    \rs
          4\cs\logms2Substract line 3 from line 2
                                  \dotfill\cs\hfill\underbar{ 800}\rs
          5\cs Add lines 1 and 4. Enter here                      \rs
           \cs\logms2and on Form 1040, line 35 \dotfill\cs\$4,800}
\def\colsepsurround{\kern.5ex}\cvsize=3ex
$$\fll\btable\data$$
```

Example *(Spreadsheet digression)*

It happens that tables—read budgets—set by (LA)TEX suffer from a simple addition flaw. Can't we add a *little* spreadsheet functionality to TEX, i.e., let TEX do the additions?[43]

[42] Note the invariance of the `\data` specification.

[43] Courtesy Phil Taylor's Trigonometry TEX, for the idea to use dimen vars and how to remove the pt. Remember that TEX allows 16 bit fixed-point arithmetic. It is limited to absolute values < 16384.

```
\def\data{r1\cs     \ass\a{1.23} \rs
          r2\cs\ubp{\ass\b{4.56}}\rs
 \cs\cs\sumup{\a\b}\global\c\dimen0\rs
 \cs\cs\ubm{\ass\a{3.14}}\rs
              \cs\cs\minus\c\a}
$$\btable\data$$
```

r1	1.23	
r2	$\underline{4.56}_{+}$	
		5.79
		$\underline{3.14}_{-}$
		2.65

with auxiliaries

```
\newdimen\a\newdimen\b\newdimen\c
\def\ass#1#2{\global#1=#2pt#2}
\def\sumup#1{\global\dimen0=0pt\let\process\processsum\fifo#1\ofif}
\def\processsum#1{\global\advance\dimen0#1\relax}
\def\ofif#1\fifo{\fi\nodim\the\dimen0}
\def\minus#1#2{\global\dimen0#1\global\advance\dimen0-#2\nodim\the\dimen0}
%\nodim, i.e. \xs, strips pt from dimension quantity
{\catcode`\p=12 \catcode`\t=12 \gdef\xs#1pt{#1}}\def\nodim{\expandafter\xs}
%\underbar extended by + and -
\def\ubp#1{\underbar{#1}\rlap{\lower.8ex\hbox{\thinspace\small$+$}}}
\def\ubm#1{\underbar{#1}\rlap{\lower.8ex\hbox{\thinspace\small$-$}}}
```

Example *(Part of AAP's table, AAP 1989)*

Subtle is the header part and the alignments on the signs.

AAP's table: Job Changes 1973–1980

	Gain/Loss of hospitals since 1973	Total No of CEO Job Changes 1973–90	Survival Rate of CEO's
Texas	+20	—	22%
Maryland	+ 5	42	24%

Source: David Kinzer, 'Turnover etc.

obtained via

```
\def\btablecaption{AAP's table:\qquad Job Changes
                               1973--1980\hfill}
\def\rowstblst{{Texas}{Maryland}}
\def\header{\copy1\cs\copy2\cs\copy3}
\def\data{$+20$\cs---\cs22\%\rs
          $+?5$\cs 42\cs24\%}
\def\footer{Source: David Kinzer, `Turnover etc.}
$$\btable\data$$
```

with auxiliaries

```
\def\nl{\hfil\strut\break\strut}
 \catcode`?=\active \def?{\kern1.1ex}
 \setbox1=\vtop{\noindent\hsize14ex
    Gain/Loss of\nl hospitals \nl since 1973}
 \setbox2=\vtop{\noindent\hsize13ex
    Total No\nl of CEO Job\nl Changes\nl 1973--90}
 \setbox3=\vtop{\noindent\hsize10ex
    Survival\nl Rate of\nl CEO's}
```

Remark. With longer footnotes put the text in a `\vtop`, with `\hsize` set to `\thsize`.[44]

[44] The table proper is first set in a box, and this box is measured yielding `\thsize` and `\tvsize`. After this the `\header` and `\footer` are set using the `\thsize`.

Example *(AT&T table diversions, The TEXbook 247)*

Subtle is the alignment on the dot via the use of '?', à la Knuth, and the `\llap\$`, and `\rlap*`.

AT&T Common Stock

Year	Price	Dividend
1971	41–54	$2.60
2	41–54	2.70
3	46–55	2.87
4	40–53	3.24
5	45–52	3.40
6	51–59	.95*

* (first quarter only)

AT&T Common Stock		
Year	Price	Dividend
1971	41–54	$2.60
2	41–54	2.70
3	46–55	2.87
4	40–53	3.24
5	45–52	3.40
6	51–59	.95*

* (first quarter only)

via

```
\def\btablecaption{AT\&T Common Stock}
\def\header{Year\cs Price\cs Dividend}
\catcode`?=\active \def?{\kern1.1ex}
\def\data{1971\cs41--54\cs\llap\$2.60\rs
             2\cs41--54\cs       2.70\rs
             3\cs46--55\cs       2.87\rs
             4\cs40--53\cs       3.24\rs
             5\cs45--52\cs       3.40\rs
             6\cs51--59\cs       ?.95\rlap*}
\def\footer{* (first quarter only)}
$$\framed\vcenter{\vbox{\small
  \btable\data}}\qquad\qquad
%
\def\btablecaption{}
\def\header{\logms3
   \hfil AT\&T Common Stock\hfil\rs
   \hfill Year\hfill\cs
   \hfill Price\hfill\cs
   \hfill Dividend\hfill}
\vcenter{\flr\ruled\btable\data}$$
```

Example *(Pittman's deterministic multiplication table)*

Typographically and in other programming languages a trifle. In TEX the coding matters.

1	2	3
2	4	6

×	1	2	3
1	1	2	3
2	2	4	6

×	1	2	3	4
1	1	2	3	4
2	2	4	6	8
3	3	6	9	12

via

```
\newcount\rcnt\newcount\ccnt\newcount\tnum\newcount\mrow\newcount\mcol
\rcnt1 \ccnt1 \mrow2 \mcol3
\def\rows{\global\ccnt1 \cols \global\advance\rcnt1
   \ifnum\rcnt>\mrow\swor\fi\rs\rows}
\def\swor#1\rows{\global\rcnt1 \global\ccnt1 \fi}
\def\cols{\tnum\rcnt\multiply\tnum\ccnt\the\tnum\global\advance\ccnt1
   \ifnum\ccnt>\mcol\sloc\fi\cs\cols}           \def\sloc#1\cols{\fi}
$$\mrow2 \mcol3 \vcenter{\btable\rows}\qquad\qquad
%
```

```
\def\first{$\times$}  \def\header{\row}
\def\row{\the\ccnt\global\advance\ccnt1 \ifnum\ccnt>\mcol\wor\fi\cs\row}
\def\wor#1\row{\fi} %\def\rowstblst{{1}{2}}%Fully automated via the two defs
   \def\rowstblst{{Anything, non-empty}}    \def\nxtrs{\the\rcnt\rss}
\vcenter{\btable\rows}\qquad\qquad
%
\mrow3 \mcol4 %\def\rowstblst{{1}{2}{3}}%Fully automated via the two defs
   \def\rowstblst{{Anything, non-empty}}     \def\nxtrs{\the\rcnt\rss}
\framed\vcenter{\btable\rows}$$
```

Kaleidoscope

Some examples of special tables are given in order to illustrate the diversity. The markup is not restricted to the use of `\btable`. With respect to programming, the deterministic tables and the tables which update the computer memory, deserve special attention. The latter category is induced by *computer* typography.

Example *(Young tableaux, irregular shape, Knuth 1973)*

```
\newdimen\csize\csize=3ex
\newcount\cnt
\def\young#1{\hbox{\vrule\vtop{\hrule
   \offinterlineskip\halign{&\vbox
   {\hbox to\csize{\strut\hss##\hss
   \vrule}\hrule}\cr%template
#1\crcr}}}}%end \young
\catcode`'=13\relax
\def'{\global\advance\cnt1 \the\cnt}
\def\data{'&'&'&'\cr'&'&'\cr'}
\young\data
\def\data{7&8&9&10\cr9&11\cr16\cr}
\young\data
```

1	2	3	4
5	6	7	
8			

7	8	9	10
9	11		
16			

Example *(Pascal triangle, a deterministic table)*

```
$$\displaylines{1\cr
  1\quad1\cr
  1\quad2\quad1\cr
  1\quad3\quad3\quad1\cr}$$
```

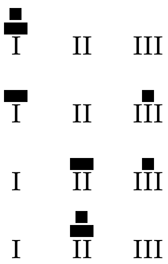
$$\displaylines{1\cr 1\quad1\cr 1\quad2\quad1\cr 1\quad3\quad3\quad1\cr}$$

4.3 The Pascal triangle is deterministic. Write a macro `\def\pascal#1{...}` which provides the table with the order as argument.

Example *(Tower of Hanoi, deterministic)*

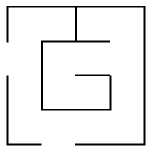
The process of replacement of the disks will be printed by the invocation of `\hanoi`$\langle n\rangle$, n an integer. No user markup is needed.

```
\beginhanoi
\hanoi2
\endhanoi
```

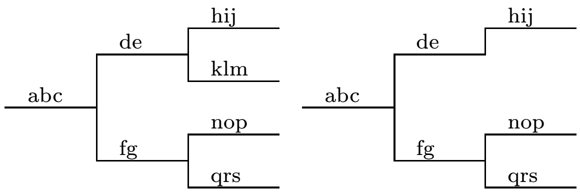


Example *(Maze)*

```
\catcode`|=13 \def|{\vrule}
\catcode`_=13 \def_{\hrulefill}
$$\vcenter{\hbox{\vtop{\hrule
\offinterlineskip
\halign{&\hbox to\csize{\vrule
  height\csize width0pt#\hfil}\cr
          |&_&|_& \cr
           &|&_&  \cr
          |&|_&_&|\cr
          |_& &_&_\cr}%end \halign
 }\vrule%end \vbox
 }%end \hbox
```



Example *(Chart, The TEXbook ex22.14)*



via the following alternative code without `\halign`

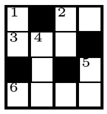
```
\newdimen\dist \dist=2\baselineskip
\def\ent#1{\hbox{\vbox to0pt{\vss\hbox to10ex{\quad
                             \strut#1\hss}\hrule}}}
\def\vl#1{\hbox{\vrule height#1\dist depth0pt}}
\def\bl#1{\kern#1\dist}%BLank of proper size
%Balanced
$$\offinterlineskip
 \vcenter{\ent{abc}}
 \vcenter{\ent{de} \vl2\ent{fg}}
 \vcenter{\ent{hij}\vl1\ent{klm}\bl1\ent{nop}\vl1\ent{qrs}}
\quad%Nearly balanced tree; pruning a branch
 \vcenter{\ent{abc}}
 \vcenter{\ent{de} \vl2\ent{fg}}
 \vcenter{\ent{hij}\vl{.5}%\bl{.5}\ent{klm}
                        \bl{1.5}\ent{nop}\vl1\ent{qrs}}$$
```

There are some special tree packages.

Example *(Much alignment occurs in typesetting mathematics)*

Aligned equations in display, matrices, and the complex commutative diagrams. This has been dealt with elsewhere, for example Knuth 1984, and Spivak 1989. Also typesetting of program texts requires vertical alignment.

Example *(Crosswords: puzzle, clues and solution)*



Across
2 Switch mode
3 Knuth
6 Prior to TEX

Down
1 Public domain
2 All right
4 All comes to it
5 Atari type

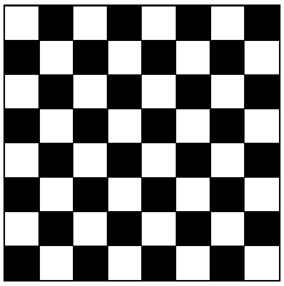
The above is obtained essentially via

```
\begincrosswords
$$\bdata%
P*On
DEk*
*n*S
Edit
\edata
\vcenter{\sevenrm\csize=3ex\crw}
⟨Clues⟩
\vcenter{\fiverm\csize=3ex\sol}$$
\endcrosswords
```

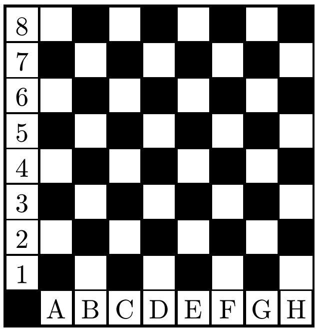
Example *(Chess boards)*

The first board has been marked up via `\leaders`,[45] horizontally and vertically. The second is obtained via (a degenerate use of :-))) the crossword macro.

```
\unitlength3ex
\boxit{\vbox{\leaders
 \vbox{\offinterlineskip
  \hbox{\leaders\hbox{%
   \kern\unitlength
   \vrule width\unitlength
          height\unitlength
  }\hskip8\unitlength
  }\hbox{\leaders\hbox{\vrule
    width\unitlength
    height\unitlength
    \kern\unitlength
  }\hskip8\unitlength
  }}\vskip8\unitlength}}
```



```
\begincrosswords
\obeyspaces\let =\space
\csize=3ex
\bdata%
8 * * * *
7* * * *
6 * * * *
5* * * *
4 * * * *
3* * * *
2 * * * *
1* * * *
*ABCDEFGH
\edata
\sol
\endcrosswords
```



4.4 How can we add field indicators on top and to the right?

4.5 How can we diminish the optical illusion of the border?

4.6 How can we add pieces to the board?

---

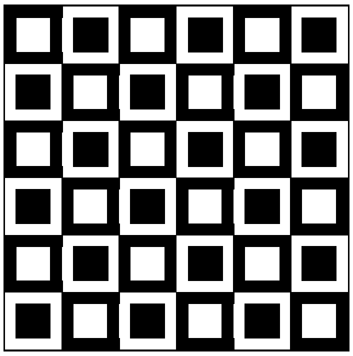
[45] Courtesy David Salomon.

The above entails that we are on the way to generate any board and just add the pieces, all based on the crosswords macro approach.

Example *(Optical illusion)*

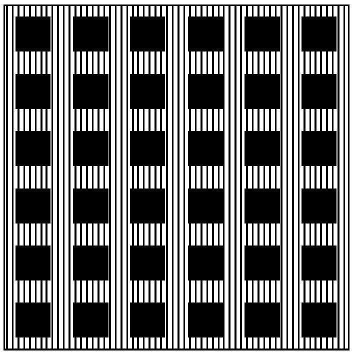
Again a nice application of the use of `\leaders`,[46] horizontally and vertically. When I submitted my crosswords macro to *TUGboat* it was refused because of the wiggling of the frame!?! There the similar effect is on with neighbouring black and white squares. The black squares look smaller than the white squares, yielding a wiggling illusion.

```
\boxit{\vbox{\leaders\vbox{%
  \offinterlineskip\hbox{\leaders
   \hbox{\copy\soto\copy\sot}%
   \hskip30ex}%
  \hbox{\leaders
   \hbox{\copy\sot\copy\soto}%
   \hskip30ex}}%
  \vskip30ex}}
```



Example *(Soto's kinetic art)*

```
\leavevmode
%background
\hbox{\xleaders\hbox to.5ex{\hss
   \vrule height6\unitlength\hss
   }\hskip6\unitlength
   }\kern-6\unitlength
%the squares
\boxit{\vbox{\offinterlineskip
   \leaders\hbox{\leaders\copy\sot
      \hskip6\unitlength
   }\vskip6\unitlength}}
```



Example *(Row and/or column spans, partial horizontal rules)*

Pair No	Contract		Re-sults	Scores		MP s
	N–S	E–W		N–S	E–W	
1.						
2.						
et cetera						

```
\newcount\bcnt
\def\lines{\global\advance\bcnt1 \ifnum\bcnt=3 et cetera\hidewidth
   \senil\fi\the\bcnt.\cs\cs\cs\cs\cs\cs\rs\lines}
\def\senil#1\lines{\fi}
\thisbtable{%
\def\ghostrow{\omit\colsepsurround%
  \vbox to0pt{\vss\hbox to5ex{\hss Pair\hss}\vskip.5ex
          \hbox to 5ex{\hss No\hss}\vss}\prs\logms2\lineglue\srp
  \vbox to0pt{\vss\hbox to5ex{\hss Re-\hss}\vskip.5ex
          \hbox to5ex{\hss sults\hss}\vss}\prs\logms2\lineglue\srp
  \vtop to0pt{\vss\hbox to5ex{\hss MP$\,$s\hss}\vss}\colsepsurround\cr}
\def\header{\cs\logms2\hfil Contract\hfil\cs
                          \cs\logms2\hfil Scores\hfil\cs\grs
            \cs N--S\cs E--W\cs\cs N--S\cs E--W\cs}
}
```

[46] For `\soto` and and `\sot` see the script file.

```
$$ \framed\ruled\btable\lines$$
```

4.7 How many rows does the header have?

4.8 The above recursive markup of the data is elegant, especially when the table is large. Simplify the markup by abandoning the independence of size, i.e., provide the data explicitly instead of recursively.

Example *(Bridge pair match scheme, with special 'first' element)*

Game Round	1–3	4–6	7–9	10–12	13–15	16–18	19–21
1.	1–2			2: 3–4		3: 5–6	4: 7–8
2.	4: 8–3	1: 1–6			2: 5–7		3: 2–4
et cetera							

via

```
\def\first{\thispicture{\unitlength=6ex\xdim{1.4}\ydim{1}}%
  \lower1.5ex\hbox\beginpicture%(1.5, 1)(0,-.1)
  \put( 0.0, 0.0){\small \strut Round}%
  \put( 1.4, 0.575){\llap{\small Game}}%
  \put( 1.4, 0.1){\line(-2, 1){1.4}}%
\endpicture
\lower2ex\null
}
\def\header{%
  1--3 \cs 4--6 \cs 7--9 \cs 10--12\cs13--15\cs 16--18 \cs 19--21}
\def\rowstblst{{1.}{2.}{et cetera}}
\def\data{%
1--2   \cs       \cs     \cs2: 3--4\cs        \cs3: 5--6 \cs4: 7--8\rs
4: 8--3\cs1: 1--6\cs     \cs       \cs2: 5--7 \cs        \cs3: 2--4\rs
       %\cs       \cs     \cs       \cs        \cs        \cs      \rs
}
\framed\ruled\btable\data
```

Remark. The need for centering the first row is induced by `\first`. It is elegant to solve the problem for that in `\first`.

4.9 How can we center the numbers in the first column?

Example *(Alignment at decimal points)*

Generally this is done by introducing an extra column for the dot, and flushing right the digits left and flushing left the digits right of it. It comes from the wish not to print non-significant leading zeroes, and to suppress zeroes behind the point for exact numbers. 3.5 means exactly $3\frac{1}{2}$. 3.500 means accurate to three digits. I would mark up the numbers separately and insert Knuth's '?', *The TEXbook* 240–241, for non-signicant zeroes, automatically. The latter can also be done by programming the editor by a template consisting of sufficient ?-s followed by the decimal point. The numbers required can be brought in by overtyping the template. Related to alignment at decimal points is alignment at number signs,[47] see the AAP table.

Example *(Nested table)*

A ruled table with unruled tables as data is incorporated in the chapter 'Creating an index,' to illustrate the ordering choices. See the script.

[47] Another column for the signs offends the structure. At a lower level one could think of `\scs`, a Sign Column Separator, but that is not nice either.

Example *(Nested, and interrupted table with updating of memory)*

From the user level one would say: aligned material connected by text, where the latter in the case at hand is the status of the play. For the markup see the script.

Puzzle	♠ KQ76	6NT,
	♡ J98	by East
	◇ J942	
	♣ 65	
♠ AJ3	N	♠ T9
♡ K653	W E	♡ A2
◇ AK3	S	◇ T5
♣ AQT		♣ KJ9xxxx
	♠ 8542	
	♡ QT74	
	◇ Q876	
	♣ 2	

Trick					NS	EW
1	♡4!	♡K	♡8	♡2	–	1
2	♣A	♣5	♣x	♣2	–	2
3	♣Q	♣6	♣x	♠2	–	3
4	♣T	♡9	♣K	♠4	–	4
5	♣J	♠5	♠3	♠6	–	5
6	♣9	♠8	♡5	♠7	–	6
7	♣x	◇6	♠J	◇2	–	7

Puzzle	♠ KQ	NS squeezed on
	♡ J	♣ continuation?
	◇ J94	
	♣ –	
♠ A	N	♠ T9
♡ 63	W E	♡ A
◇ AK3	S	◇ T5
♣ –		♣ ⓧ
	♠ –	
	♡ QT7	
	◇ Q87	
	♣ –	

8	♣x	♡7	♡6	♡J	–	8
et cetera						

Tables which also require updating of memory occur with typesetting of chess, for example Tutelaers 1992, or GO, Kołodzjieska 1991.

Example *(Fill-in forms)*

The registration forms for the various TEX conferences are captivating. Now and then I ponder about the e-mail equivalents of the traditional snail forms, especially what (and how they) should be filled in. Redefine a list of empty definitions?

Example *(Time tables)*

Railway, bus, . . . , and the use of rotated fonts. This is a PostScript issue.

Example *(No markup)*

For simple tables the data can be specified *without* `\cs`-s, and `\rs`-s, i.e., without explicit markers for column and row separators. These can be inserted by TEX, see *The TEXbook* 249. For special cases, for example centering texts[48] and crosswords, this is handy.

Pondering about lists makes you realize that the table proper data forms a nested list with `\rs` and `\cs`, respectively first level and second level separators. One could also adopt ⟨*cr*⟩ and ␣ for that, in other words allow natural data provision.

[48] See the implementation of `\center` in the file.

Example *(Simultaneous row spans)*

Simultaneous column spans can be marked up in `\btable` by `\logms`—for logical columns, as an extension to `\multispan` for `\halign`. For row spans start from `\valign`, the transposition of `\halign`. A little more than the analogy of transposition is needed. From the TEX-nl list the following example of spans over rows.

```
\hbox{\vrule\valign{&\hrule\vfil
  \noindent\hsize15ex\strut
  \xsp#\xsp\vfil\hrule\cr
\multispan3{\hrule\vfil
  \vbox{\noindent\hsize15ex\strut
   \xsp This\\
   \xsp looks\\
   \xsp problematic}}\vfil
   \hrule&
   Doesn't it?\cr
\noalign{\vrule}
\hfill22 &%space needed
\hfill12 &
\hfill1943 &
\hfill190 \cr}\vrule}
%with auxiliary
\def\xsp{\hbox{ }}
```

<table>
<tr><td rowspan="3">This
looks
problematic</td><td align="right">22</td></tr>
<tr><td align="right">12</td></tr>
<tr><td align="right">1943</td></tr>
<tr><td>Doesn't it?</td><td align="right">190</td></tr>
</table>

As can be seen from this difficult example, we have to fall back on lower level details now and then.

4.10 Modify the above towards a `\vtable` macro, such that we can use the attributes `\ruled`, `\vruled`, `\hruled`, with default `\nonruled`, and `\framed` and `\nonframed`.

Example *(Macroscopic markup vtable, based on valign)*

```
\def\data{\logms3{\hsize15ex
   \vfil\noindent\xsp This\\
                 \xsp looks\\
                 \xsp problematic
   \vfil}\rs\omit\hsize15ex
   \vfil\noindent
         \xsp Doesn't it?\vfil\cs
          O\rs  A\rs G\rs 189\cs
           \rs 16\rs11\rs1954\lcs}
\framed\ruled\vtable{5ex}
```

<table>
<tr><td rowspan="3">This
looks
problematic</td><td>O</td><td></td></tr>
<tr><td>A</td><td>16</td></tr>
<tr><td>G</td><td>11</td></tr>
<tr><td>Doesn't it?</td><td></td><td>1954</td></tr>
</table>

Example *(Simultaneous row and column spans)*

These require more work.

Caption

<table>
<tr><td></td><td colspan="3">Header</td><td>⇓</td></tr>
<tr><td>row 1</td><td colspan="3" rowspan="2">$2 * 3$</td><td>14</td></tr>
<tr><td>row 2</td><td>24</td></tr>
<tr><td>⇒</td><td>31</td><td>32</td><td>33</td><td>34</td></tr>
</table>

Footer

Caption

<table>
<tr><td></td><td colspan="3">Header</td><td>⇓</td></tr>
<tr><td>row 1</td><td colspan="3" rowspan="3">$3 * 3$</td><td>14</td></tr>
<tr><td>row 2</td><td>24</td></tr>
<tr><td>row 3</td><td>34</td></tr>
<tr><td>⇒</td><td>41</td><td>42</td><td>43</td><td>44</td></tr>
</table>

Footer

4.11 How to markup the above with simultaneous row and column spans?

Example *(Gnonoms)*

In *TUGboat* a discussion about the following kind of rules within tables—so-called gnonoms—emerged.

```
\newcount\columnno \columnno1
\def\data{1\cs3\cs5\rs
          1\cs4\cs7\rs
          1\cs5\cs9\lrs}
\def\cs{\hss\vrule\ifnum
   \columnno<\rowno width0pt\fi\relax
         &\global\advance\columnno1 }
\def\rs{\hss\vrule\relax\cr
        \global\advance\rowno1
        \global\columnno1 }
\def\lrs{\hss\vrule\relax\cr}
\halign{&\vbox to12pt{\hbox to12pt
  {\hss\strut#}\vss\hrule\ifnum
 \rowno<\columnno width0pt\fi}\cr
        \data}
```

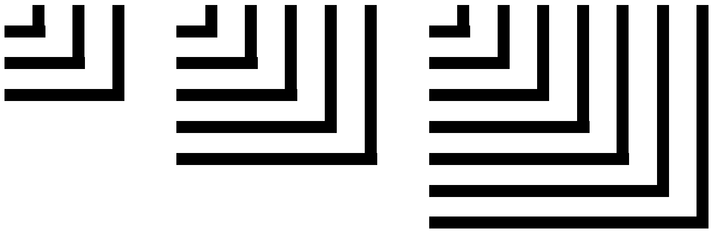
1	3	5
1	4	7
1	5	9

4.12 Each entry of the table obeys the formula

$$entry_{i,j} = 2j - 1 + (i-1)(j-1).$$

A little backside of the envelope calculation—-or just look closer at the table—shows that elements in neighbouring columns differ by $i + 1$. Format this table as function of the size, i.e., with the elements generated automatically.

4.13 Playing longer with the gnonoms can yield the following graphical results. Modify the above in order to obtain this graph.



Font charts Font tables can be formatted via `\beginchart` and `\endchart`. This markup is used in *The TEXbook* 468–526, for the font tables of the Appendices C and F, and in `testfont.tex`, which comes along with METAFONT.

Example *(ASCII table, The TEXbook 367)*

```
\beginchart{\global\count@='41\tentt
 \def\chartstrut{\lower4.3pt\vbox to13.6pt{}}}
&\oct{00x}&&NUL&&SOH&&STX&&ETX&&EOT&&ENQ&&ACK&&BEL&&\oddline0
&\oct{03x}&&CAN&&EM&&SUB&&ESC&&FS&&GS&&RS&&US&\evenline
&\oct{04x}&&SP&&\:&&\:&&\:&&\:&&\:&&\:&&\:&&\oddline2
&\oct{05x}&&\:&&\:&&\:&&\:&&\:&&\:&&\:&&\:&\evenline
\noalign{\vskip2ex\line{\hfil et cetera\hfil}
         \vskip2.5ex\hrule}
&\oct{16x}&&\:&&\:&&\:&&\:&&\:&&\:&&\:&&\:&&\oddline7
&\oct{17x}&&\:&&\:&&\:&&\:&&\:&&\:&&\:&&DEL&\evenline
\endchart
```

resulting in the partial ASCII table

	´*0*	´*1*	´*2*	´*3*	´*4*	´*5*	´*6*	´*7*	
´*00x*	`NUL`	`SOH`	`STX`	`ETX`	`EOT`	`ENQ`	`ACK`	`BEL`	″0x
´*03x*	`CAN`	`EM`	`SUB`	`ESC`	`FS`	`GS`	`RS`	`US`	
´*04x*	`SP`	`!`	`"`	`#`	`$`	`%`	`&`	`'`	″2x
´*05x*	`(`	`)`	`*`	`+`	`,`	`-`	`.`	`/`	

`et cetera`

´*16x*	`0`	`1`	`2`	`3`	`4`	`5`	`6`	`7`	″7x
´*17x*	`8`	`9`	`:`	`;`	`<`	`=`	`>`	`DEL`	
	″8	″9	″A	″B	″C	″D	″E	″F	

Example *(Markup for Appendix F 427, 428, 429, 430, 431, 432)*

```
\beginchart{\postdisplaypenalty=0 \tenrm}
                                    %or \tenit, \tentt, \tenit, \tensy, \tenex
      \normalchart
\endchart
```

resulting in, for the font chart `\tenrm`

	´*0*	´*1*	´*2*	´*3*	´*4*	´*5*	´*6*	´*7*	
´*00x*	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	″0x
´*01x*	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
´*02x*	ı	ȷ	`	´	ˇ	˘	¯	˚	″1x
´*03x*	¸	ß	æ	œ	ø	Æ	Œ	Ø	
´*04x*	-	!	”	#	$	%	&	’	″2x
´*05x*	(	)	*	+	,	-	.	/	
´*06x*	0	1	2	3	4	5	6	7	″3x
´*07x*	8	9	:	;	¡	=	¿	?	
´*10x*	@	A	B	C	D	E	F	G	″4x
´*11x*	H	I	J	K	L	M	N	O	
´*12x*	P	Q	R	S	T	U	V	W	″5x
´*13x*	X	Y	Z	[	“	]	ˆ	˙	
´*14x*	‘	a	b	c	d	e	f	g	″6x
´*15x*	h	i	j	k	l	m	n	o	
´*16x*	p	q	r	s	t	u	v	w	″7x
´*17x*	x	y	z	–	—	˝	˜	¨	
	″8	″9	″A	″B	″C	″D	″E	″F	

The default value of `\postdisplaypenalty` is -10000, to enforce a page break.

4.14 How can we get a print of LATEX's line fonts? Why don't you print `\oldstyle` too, just for the fun of it?

Why bother about font charts? Well, ... a lot can be learned from `\beginchart` for the markup of classes of tables.

- Font charts belong to the class of bordered tables
- the template is hidden from the user
- the data can be provided separately and processed on the fly
- the default value of `\displaypenalty` can be redefined via `\displaypenalty` ⟨*number*⟩ as argument of `\beginchart`

- `\beginchart` is parameterized over the fonts
- `\normalchart` keeps automatically track of the positions in the font chart.

In short, the macros show how

- to mark up classes of tables
- to hide an `\halign` template
- to use two-part macros
- to cope with options and defaults
- to parameterize over fonts, and
- to minimize markup.

The argument of `\beginchart` is used among other things to control the kind of table: `\tenrm`, `\tenit`, `\tentt`, `\teni`, `\tensy`, or `\tenex`. The data for normal font tables is provided via `\def\normalchart{...}`. As a consequence the markup for the other tables in *The TEXbook* Appendix F is surprisingly simple.

4.15 How come that Knuth decided to implement an ordinary argument after `\beginchart`? If you look at `\begindisplay` nearly the same coding structure is used, but there the argument is optional. Subtle, very subtle these coding schemes.

Pictures

blue.tex contains graphics facilities collected from The TEXbook, manmac and gkpmac. The gkpmac picture macros provide a subset of LaTEX's functionality. I introduced a database of pictures to cooperate with TEX. Two main issues have been addressed: using a picture from the picturebase, and creating a picture. No interaction with METAFONT (or METApost) nor the incorporation of encapsulated PostScript have been treated.

blue.tex allows pictures to be used from the picture database pic.dat. For creating pictures manmac's `\makeblankbox`, and gkpmac's picture environment are incorporated.[49]

The roots

In *The TEXbook* Appendix D, Knuth states

> 'TEX is designed to put boxes together either horizontally or vertically, not diagonally. But that is not a serious limitation, because the use of negative spacing makes it possible to put things anywhere on the page.'

The following, from *The TEXbook* 389, is the basis for LATEX's picture environment.

```
\hbox{\unit\baselineskip\point 0 0 \point 1 1.5 }%etc.
%with
\newdimen\unit
\def\point#1 #2 {\rlap{\kern#1\unit\raise#2\unit\hbox{$
   \scriptstyle\bullet\;(#1,#2)$}}}
```

The examples with quarter circles, *The TEXbook* 389–390, the dragon figures, remind me of Papert's 'Turtle' graphics.

Using pictures

pic.dat contains definitions of prefab pictures. In order to use these, the picture must be loaded via

`\beginpictures`
`\`⟨*name*$_1$⟩`pic`...`\`⟨*name*$_n$⟩`pic`   or   `\pictures{\`⟨*name*$_1$⟩`pic`...`\`⟨*name*$_n$⟩`pic}`
`\endpictures`

After this each picture loaded can be used by its name, i.e., via the invocation

`\`⟨*name*$_x$⟩`pic`

if necessary preceded by `\thispicture{...}`, with actual values for `\unitlength`, and the offsets. The result is packed as an `\hbox to#1\unitlength{...}` with depth reset to zero.

Note that although the picture database is a file of defs, alphabetically sorted on definition name, the names of the pictures to be selected can be supplied in any order to `\pictures`.

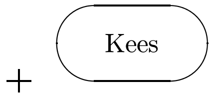
5.1 What is the effect of `\pictures{}`?

---

49 METAFONT or METAPOST are not part of blue.tex, nor is the use of encapsulated POSTSCRIPT considered. BLUe's format approach entails a high degree of portability at the expense of creating picturebase entries for reuse, which can be tedious.

Example *(Simple picture)*

```
\thispicture{\markorigintrue}
\simplepic
```



Creating pictures

I'll discuss two approaches: one for simple schemes, and the picture environment as used by Graham, Knuth, and Pastashnik.

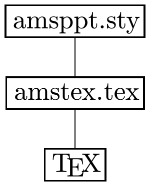
Stacking boxes For simple schemes we can use framed boxes. blue.tex provides

```
\def\fbox#1{\setbox0\hbox{\strut$\;$#1$\,$}\leavevmode
 \rlap{\copy0}\makeblankbox{.2pt}{.2pt}}
```

The idea is that the argument is set in box 0. This box is framed by manmac's `\makeblankbox`.

Example *(Simple scheme)*

```
\stack{\fbox{amsppt.sty}}
\vconnector
\stack{\fbox{amstex.tex}}
\vconnector
\stack{\fbox{\TeX}}}
%with
\def\stack#1{\hbox to10ex{\hss
                       #1\hss}}
\def\vconnector{\stack{\vrule
   height2.5exdepth1exwidth.2pt}}
```

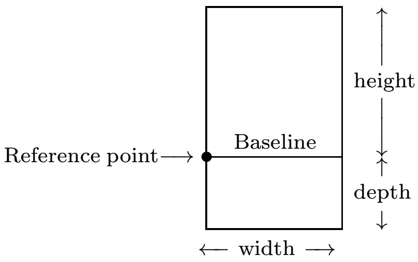


Samplebox diversions A lot of details can be learned from Knuth's markup of the `\samplebox` picture, *The TEXbook* chapter 11: composing long vertical arrows, to measure the axis height, samplebox coding. For daily use this markup is too cumbersome. However, the `\samplebox` macro is amply used in *The TEXbook* chapter 12.

Example *(Manmanc's samplebox)*

Knuth attaches three dimensions to these boxes: height, width, and depth.

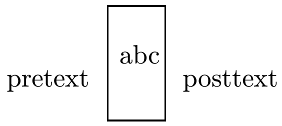
```
%Vertical arrows, similar to
%\arrows, but now vertically.
\setbox0=\hbox{$\uparrow$}
\setbox1=\hbox to \wd0{$\hss\mid
\hss$}%with luck,they'll line up
\def\vline{\nointerlineskip
  \kern-.5pt\copy1}
\setbox2=\vbox{\copy0
  \vline\vline
  \moveleft 1em\hbox{height}
  \copy1\vline
  \nointerlineskip\kern-.5pt
  \hbox{$\downarrow$}\kern.2pt}
%To measure axis height
%\ht0 is the axis height
\setbox0=\hbox{$\vcenter{}$}
```



From the above I distilled the following `\gbox` macro—generalized samplebox—which draws a box similar to sample box, with depth, height and width, and a hook to put in material relative to the reference point. This is a generalization of a framed box, because the frame has its own dimensions independent of the material to be framed. In order to be in full control of the positioning of the elements, I chose the parameters `\xshift` and `\yshift`—dimen vars—to shift material relative to the reference point.

Example *(Generalized samplebox macro)*

```
pretext
\xshift1ex \yshift2ex
\gbox{7ex}{3ex}{5ex}{abc}
posttext
%with the generalized sample box
\def\gbox#1#2#3#4{%#1=ht,#2=dp
%#3=wd, #4=text.
%Assumed: the dimen vars \xshift,
%\yshift have been initialized.
 {\setbox0=\vtop{\vbox to#1{%
           \hbox to#3{}\vss}
  \nointerlineskip
  \vbox to#2{}}
%now \box0 has the desired
%ht, dp, and wd
 \hbox{\makelightbox\kern-#3%
  \raise\yshift\hbox to#3{%
        \kern\xshift#4\hss}}}}
```



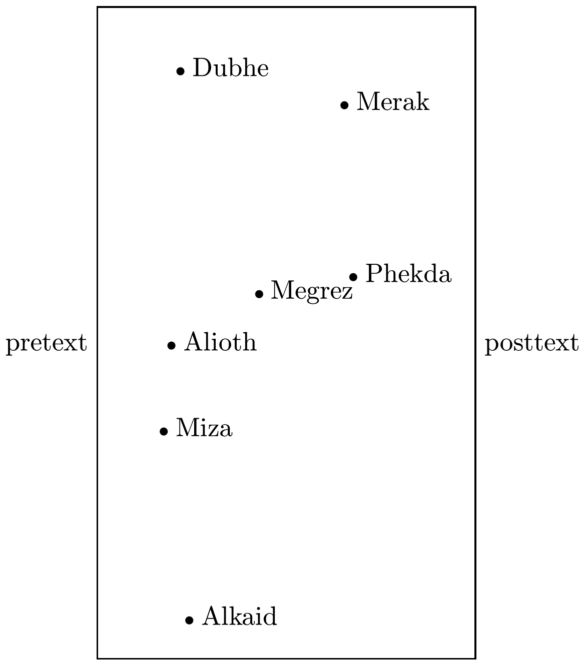
The markup of the picture of astronomy data, *The TEXbook* 389, can be simplified. Also `\point` can be generalized as follows.

```
\def\point#1 #2 #3{\kern#1\unit\raise#2\unit\hbox to0pt{\the\prefil
   \the\prepoint#3\the\postfil}\kern-#1\unit}
%with parameters
\prepoint to allow material to be inserted (toks var)[50]
\prefil   to insert the fill              (toks var)
\postfil  to insert the fill              (toks var, default \hss)
```

Example *(Atlas The TEXbook 389)*

Use `\gbox` of the right size with the data in a box.

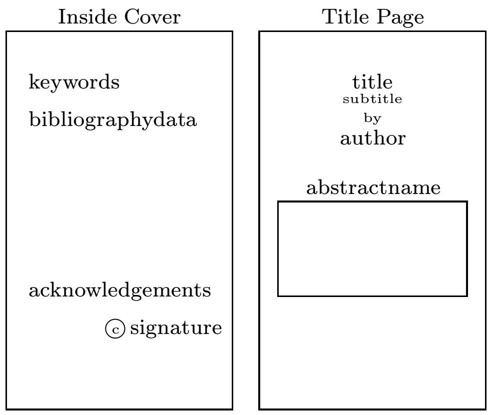
```
\unit3ex
\prepoint{$\scriptstyle\bullet\;$}
\setbox1=
\hbox{\point 0 0 {Alioth}
      \point 0 8 {Dubhe}
      \point 0 -8 {Alkaid}
      \point -1 -2.5 {Miza}
      \point 4 7 {Merak}
      \point 4 2 {Phekda}
      \point 1 1.5 {Megrez}
}
\xshift2\unit
pretext \gbox{10\unit}%height
             {9\unit} %depth
             {11\unit}%width
             {\copy1} %material
posttext
```



Example *(Layout pages)*

`\gbox` and `\point` can be combined for drawing a rough design of pages, what the pages should look like. As an example a sketch of the Inside Cover Page and the Title Page as implemented in the `\report` format has been given below.

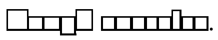
```
\unit1cm
\prepoint{\small}
\noindent\setbox1=\hbox{%
\point1.3 1 {{\tiny\copyright}%
         \thinspace signature}%
\point.3 1.5 {acknowledgements}%
\point.3 3.75 {bibliographydata}%
\point.3 4.25 {keywords}%
{\prefil{\hss}\point1.5 5.1
            {Inside Cover}}}
\gbox{5\unit} %height
     {0pt}    %depth
     {3\unit} %width
     {\copy1} %material
%
\prefil{\hss}
\setbox1=\hbox{%
\point1.5 2.85 {abstractname}%
\point1.5 1.5 {\gbox{1.25\unit}
            {0pt}{2.5\unit}{}}%
\point1.5 3.5 {author}%
\point1.5 4.05 {\tiny subtitle}%
\point1.5 3.8 {\tiny by}%
\point1.5 4.25 {title}%
\point1.5 5.1 {Title Page}}
\gbox{5\unit} %height
     {0pt}    %depth
     {3\unit} %width
     {\copy1} %material
```



5.2 How can we draw curves, for example a spline relative to the reference point?

Example *(Typesetting outlines, The TeXbook ex11.5)*
Just an outline of characters is also used in the logo of the GUST bulletin. Below is my solution to *The TeXbook* ex11.5—via nested FIFO, processing: words, via `\fifow`, and characters, via `\fifo`—to this 'tough problem.'

```
\fifow Tough  exercise
       \wofif \unskip.
%with
\def\fifow#1 {\ifx\wofif#1\wofif\fi
   \processw{#1}\fifow}
\def\processw#1{\fifo#1 \ofif\ }
\def\process#1{\boxit#1}
\def\boxit#1{\setbox0=\hbox{#1}%
  \hbox{\lower\dp0\vbox{\hrule
   \hbox{\vrule\phantom#1\vrule}%
   \hrule}}}
```



Picture environment It has been called in the file of gkpmac 'Pictures (a subset of LaTeX's conventions).' LaTeX's line fonts are used. Macros are provided for lines, vectors, curves, disks, circles, and ovals. The lines and vectors have discrete orientations and can vary in thickness (2 states). The implementation is simpler than LaTeX's, because the options have been omitted. Dimensionless units get their value via `\unitlength`. The documentation of the picture environment in pseudo-code comes with LaTeX, and applies to this set too. In blue.tex a macro for rhombus-framed elements has been added to the facilities provided by gkpmac.

The markup for the environment reads `\gkpbeginpicture` and `\gkpendpicture`.[51]

```
\def\gkpbeginpicture(#1,#2)(#3,#4){%#1,#2 x,y-dimensions; #3,#4 x,y-offsets
...}
```

Line thickness Default is `\thinlines` which can be toggled via the use of `\thicklines`. This does not apply to curves.

(multi)put These put control sequences are used for the positioning of picture objects.

```
\def\put(#1,#2)#3{%#1,#2 x,y-coordinates
   %#3 picture object, printed at #1,#2 to the right
   \raise#2\unitlength
     \rlap{\kern#1\unitlength #3}\ignorespaces}
\def\multiput(#1,#2)(#3,#4)#5#6{%
%#1,#2 x,y-coordinates; #3,#4 x,y-increases;
%#5 number of times; #6 picture object
...}
```

Picture objects Provided are: (h)box, line, vector, disk, circle, oval, and rhombus.

```
\def\makebox(#1,#2)#3{%#1,#2 x,y-sizes (dimensionless)
%#3 is put in hbox to#1\unitlength{\hss#3\hss}
...}
%
\def\line(#1,#2)#3{%abs(#1,#2)<=6
%#1,#2 horizontal,vertical-slopes; #3 horizontal size
...}
%
\def\vector(#1,#2)#3{%abs(#1,#2)<=4
%#1,#2 horizontal, vertical slopes; #3 horizontal size
...}
%
\def\disk#1{%#1 diameter <=.2inch
...}
%
\def\circle#1{%#1 diameter <=.5inch
...}
%
\def\oval(#1,#2){%#1,#2 x,y-sizes (dimensionless)
                %centre put at reference point
...}
\def\rhombus(#1,#2)#3#4{%#1,#2 hor. respectively vert. ratio <=6
   %#3 the size in unitlengths (half of total width)
   %#4 (text) put in the middle, that is the reference point.
...}
```

Example *(Simple picture markup)*

```
\unitlength=1cm
\begingkppicture(3,1.25)(0,0)
   \ifmarkorigin\put(0,0)\markorigin\fi
   \put(1.5,.5){\oval(2,1)}
   \put(0.5,0){\makebox(2,1){Kees}}
\endgkppicture
```

[51] Not conform to LaTeX, because the offset parameters are obligatory. These two outer level names have been prefixed by gkp, to reserve `\beginpicture` and `\endpicture`.

Curves The method has been explained in the METAFONT book page 13. The functionality provided is similar to Lamport's Bezier.sty, although here a quadratic spline is used.

```
\def\squine(#1,#2,#3,#4,#5,#6){%
%#1, #2, #3 x-coordinates of consecutive points
%#4, #5, #6 y-coordinates of consecutive points
%To plot f(x) between x0 and x1
%\put(0, 0){\squine(x0,xm,x1,y0,ym,y1)}, with
%y0=f(x0), y1=f(x1)
%xm=(y0-y1+s1x1-s0x0)/(s1-s0)
%ym=(s0(s1x1-y1)-s1(s0x0-y0))/(s1-s0)
%s0=f'(x0), s1=f'(x1)
```

The 'line thickness' can be varied by redefining `\squineelem`. Default: `\def\squineelem{.}`.

Adding a picture to the database pic.dat

Why a picturebase? To begin with, it is convenient to have all your pictures at hand and well-organized. Furthermore, it is nice to load the picture from the picturebase, and use it with 'arguments,' especially with the value for `\unitlength`, or actual values for the offsets, and if necessary with values for the x,y-dimensions. The defaults can be changed via providing values in `\thispicture{...}`. `\pictures` load all the specified names as defs. The above ideas resulted in the following approach.

Example *(Entry for pic.dat)*

`\unitlength` is a dimension variable, and `\xdim`, `\ydim`, `\xoffset`, and `\yoffset` are token variables. Each entry is a list element preceded by the list element tag, `\lst`, and its name.

```
\lst\simplepic{\bgroup%next supply defaults
  \unitlength=1cm\xdim{3}\ydim{1.25}%offsets are 0 by default
\beginpicture
   \ifmarkorigin\put(0,0)\markorigin\fi
   \put(1.5,.5){\oval(2,1)}
   \put(0.5,0){\makebox(2,1){Kees}}
\endpicture\egroup}%end simplepic
%with in \bluetex{} the shortcuts
\def\beginpicture{\the\everypicture\the\thispicture
 \gkpbeginpicture(\the\xdim,\the\ydim)(\the\xoffset,\the\yoffset)}
\def\endpicture{\gkpendpicture\global\thispicture{}}
```

A little more abstract an entry for pic.dat reads as follows.

```
\lst\⟨name⟩pic{\bgroup%next the defaults
 \unitlength⟨dimen value⟩\xdim{⟨number⟩}\ydim{⟨number⟩}
 \xoffset{⟨number⟩}\yoffset{⟨number⟩}
\beginpicture        %next the picture elements
   ⟨picture elements⟩
\endpicture\egroup}
```

5.3 How can we get a list of names of the pictures in the database?

Check for data integrity After a picture has been added one can check whether the file can be scanned by TEX via making a table of contents as follows.

```
\input blue.tex
\contentsdatabase{pic}
\bye
```

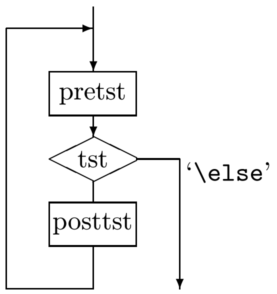
This script produces no pages of output (that is OK). The file contentspic will be created, though.

Examples

These are added to illustrate what can be achieved with the picture environment. It is true that creating the markup is not as simple as drawing by the mouse. But once you have got the template, (parts of) it can be reused. I explained the approach in broad terms.

Example *(Flow chart loop: flowchartlooppic)*

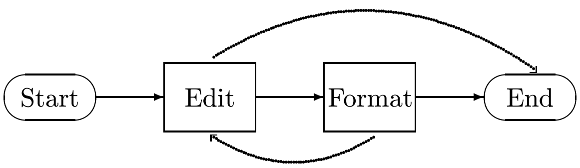
```
Interesting is the rhombus.
Similar to \oval and \framebox.
\def\rhombus(#1,#2)#3#4{%
%#1,#2 ratio of hor./vert. size, <=6
%#3 the size in unitlengths
%   (half of total width)
%#4 text to be put in the middle.
%The centre is the reference point,
%similar as with \oval.
\setbox0=\hbox{\line(#1,#2){#3}}%
\setbox1=\hbox{\line(#1,-#2){#3}}%
\hbox to0pt{\hss
   \vbox to0pt{\offinterlineskip\vss
               \copy0\copy1\vss}%
   \vbox to0pt{\vss\copy1\copy0\vss
               }\hss}%
   \vbox to0pt{\vss
    \hbox to 0pt{\hss#4\hss}\vss}}
```



Example *(Flow chart due to Furuta: furutapic)*

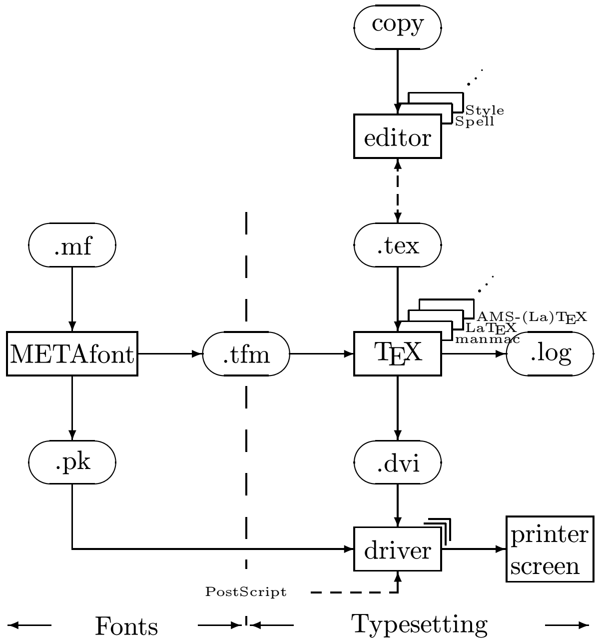
Note the use of PMA—poor man's arrow heads.

```
Interesting here are the arcs and
the arrow heads.
The latter have been provided
separately for reuse.
The reference point is at
the head.
%lu is pointing left up,
%rd is pointing right down
\def\rdarrowhead{\hbox
  to0pt{\hss\vrule height.1pt
            depth.1pt
            width.5ex
   \vrule height.5ex\relax}}
\def\luarrowhead{\hbox
  to0pt{\vrule depth.5ex
   \vrule height.1pt
          depth.1pt
          width.5ex}}
```



Example *(Salomon's overviewtexpic)*

```
The right upper corners,
suggesting multiple copies,
are nice.
Recursive use of \put has
been applied. Use
\put(){\corners{⟨size⟩}}
The reference point is ⟨x⟩,⟨y⟩.
\def\corners#1{%#1 the size
 \put(.1,.1){\line(-1,0){#1}}%
 \put(.1,.1){\line(0,-1){#1}}%
 \put(.2,.2){\line(-1,0){#1}}%
 \put(.2,.2){\line(0,-1){#1}}%
}
```



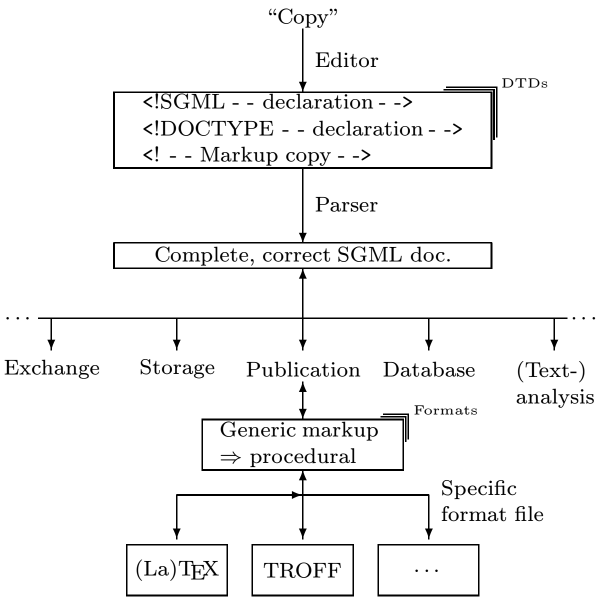
5·4 Apply the same approach for a macro `\card`, i.e., for the figure to denote multiples, as used with TEX, LATEX, AMS-(LA)TEX, to denote some of TEX's flavours.

5·5 And what about the diagonal dots?

Example *(SGML-TEX relation)*

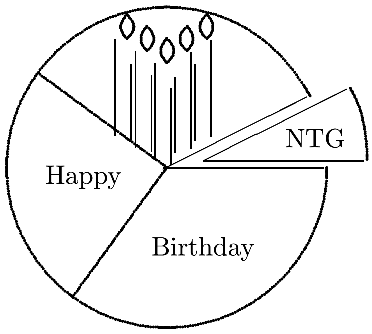
```
The interesting thing in the
markup is the abstraction of
the level in the picture via
the counter \leg.
Instead of explicit numbers
the counter is used via
\the\leg⟨.fraction⟩, and
advanced for the next level.

\thispicture{\small
    \unitlength=3ex
    \xoffset{-9}
}\sgmltexpic
```



Example *(Pie-chart: ntgpiepic)*

```
The challenge was to create a
‘circle’ via splines.
Another thing is that the top
of the candles is done by \line,
which does not show in print
when too small.
The origin is close to the middle
of the chart.
```



5.6 How to generalize the above picture in such a way that a business pie-chart can be represented?

The picture on the cover—BLUe’s format in a nutshell—is nice too. Consult lit.dat for the markup.

Coping with References

blue.tex makes use of the database of references lit.dat. The references have to be specified by name. Prior to the formatting of the script the specified references are set and kept in a box. This box is pasted up on command. The layout of the references has been inspired upon AMS and can be customized at a lower level.

Most if not all of your references should be part of your[52] database of literature. This starting point makes things simpler and definitely facilitates reuse.

blue.tex makes use of lit.dat, a database of definitions. The names of the definitions—primary keys in database jargon—identify the entries. Each replacement text contains the marked up reference. The structure of the marked up reference is modest and prompted by my use: author name, year of publication, and rest part, next to some adopted conventions. In the field I'm working I know the publication by heart, once I know the name of the author(s) and the year of publication. For my own use I don't need further tagging.

Using references

With blue.tex the use comes down to specifying the names of the references in the preliminary part of the script, via

`\beginreferences`
$\backslash\langle \mathit{name}_1\rangle\ldots\backslash\langle \mathit{name}_n\rangle$ or `\references`$\{\backslash\langle \mathit{name}_1\rangle\ldots\backslash\langle \mathit{name}_n\rangle\}$
`\endreferences`

and paste up the references via `\pasteupreferences`, at your place of choice.

6.1 What is the effect of `\references{}`?

Placement within context can be handled via the token variables `\prereferences` and `\postreferences`. Default `\prereferences` takes care of the heading via

`\prereferences{\head{\the\referencesname}}`.

The token variables `\thisreferences` and `\everyreferences` can be used for customization.

Cross-referencing can always be done via the convention of $\langle \mathit{author\ name(s)}\rangle$ $\langle \mathit{year}\rangle$. This approach facilitates reading, at least for me, at the expense of a little more space in print than the system of numbers. The latter system is possible with blue.tex. Once the references have been set by `\references`, each name is redefined by its sequence number. After `\references`$\{\ldots\backslash\langle \mathit{name}_n\rangle\ldots\}$ then $\backslash\langle \mathit{name}_n\rangle$ denotes the number $\langle n\rangle$.

Robustness is there too. Most likely users will now and then mistype a name supplied in `\references`. blue.tex reacts very elegantly to this. First, it continues, it does not stop. A message is provided in the log file, and as reference an error message is printed by `\pasteupreferences`.

[52] And hopefully of your publisher as well.

Adding a reference to the database lit.dat

The structure of the bibliography items in lit.dat is modest, a list of elements and each element starting with the list element tag, `\lst`, and a name.

```
\lst\⟨name⟩⟨extension⟩
     {⟨name part⟩ (⟨date part⟩):
      ⟨title part⟩
      ⟨rest⟩
}
```

This syntax leaves freedom and flexibility, although the essentials—name, date, title[53] — have prescribed positions. The number of authors does not matter, nor is it prescribed how the author names should be specified. That is up to you. The representation of the date is free too, leaving room for things like 'in progress' or 'priv. comm.' I don't like to prescribe structurally the title in more detail. The rest part is not further prescribed either, because of the great variety of sources—publishers, books, journal series, proceedings, theses, technical reports, and the like. At the end of the rest part I added, within parentheses, annotations, now and then.[54]

For the name I chose to use lowercase letters only: the (first) name followed by the initials. For the ⟨*extension*⟩ I chose extra letter(s). The latter are there to tell the difference between more than one publication per author, and to allow insertions, when one stumbles on some early work of an author. As an example of an entry, I'm pleased to take *The TEXbook*.[55]

Example *(lit.dat entry)*

```
\lst\knuthded
   {Knuth D.E (1984):
    \TB.
    \AW. ISBN 0-201-13448-9.
    \annotation{Soft cover edition of book A in the
     series Computers and Typesetting.}
}
```

The list of references is ordered alphabetically on ⟨*name*⟩⟨*extension*⟩. Prepare your bibliography and insert the items in your personalized version of lit.dat, and that is it.

6.2 How can we format the complete lit.dat database, with each entry preceded with numbers enclosed in square brackets?

6.3 How can we get a list of names of the references in the database?

6.4 When a reference has been added to lit.dat a simple check for verifying that the file can still be processed is

```
\input blue.tex
\references\zusek
\zusek
\bye
```

What would you expect as outcome?

53 I consider it important to delineate the main fields, however.

54 There is usually too much formatting in a bibliography for my taste. Why discriminate between a title of a book and an article? Given the context that is clear enough. Especially, when for a book the ISBN (or ISSN) number is provided. And why so much (superfluous) punctuation? For me there should be as little punctuation as is functionally needed to separate the elements within the entry.

55 In practice I hardly use this one, because in `\prereferences` I supply the note 'I assume *The TEXbook* to be omni-present.'

Check for data integrity After a reference has been added one can check whether the file can be scanned by TEX via making a table of contents as follows.

```
\input blue.tex
\contentsdatabase{lit}
\bye
```

This script produces no pages of output (that is OK). The file contentslit will be created, though. If you have a file with references of your own, say abc.dat, provide also `\newwrite \tocabc`.

Customization

Each entry is typeset by `\ls` which is activated by `\lsams` in `\references`. The default formatting has been inspired upon AMS.

How to provide your way of formatting? To illustrate how a user can specify his own convention blue.tex contains the template

```
\def\lsstraight{\let\ls\nitem}
```

Example *(Formatting references your way)*

Provide your variant of `\lsstraight` and supply

```
\thisreferences{\let\lsams\lsstraight}.
```

Searching the lit.dat database by pattern

BLUe's format provides a `\search` macro with as argument a search pattern. The purpose is to search the database file provided in the toks variable `\searchfile`, for this search pattern. The result of the search is the list of names of which the replacement texts contain the search pattern. This list of names is delivered in the toks var `\namelist`, and for the time being they are also written to the log file.

Example *(Search for references of 1973 in lit.dat)*

In order to justify my use of the word database in relation to TEX, I provided the search macro to browse the databases lit.dat and address.dat. Below the input and result have been given of the search of lit.dat for the pattern 1973. The result is contained in the toks variable `\namelst`.

A search pattern should not contain a period.

```
\input blue.tex
\searchfile{lit}
\search{1973}                 \lst\knuthdec\lst\schumacherefa\lst\wirthna
\bye
```

Creating an Index

The creation of a modest index within a one-pass TEX job has been treated. In general a proof run and a final run are needed.

Making an index is an art. The fundamental problem is

What to include in an index?

Computer-assisted indexing is not simple either. Issues are

- the markup of keywords or phrases
- to associate page numbers
- to sort and compress raw Index Reminders (IRs), and
- to typeset the result.

My approach is to create proof indexes for each chapter and learn from those what should be included in the total index. I perceived this as very pleasant in practice. Even if you prefer `\makeindex` for the real index, this processing on the fly of a chapter index can be of great help.

Use

I'll show how to mark up Knuth's four types of IRs, how to mark up accents, how to mark up font switching, and how to mark up spaces as part of the IR.

Example *(Markup, commands and resulting index)*

The right column has been obtained via

- `\loadindexmacros`, at the beginning of the script
- `\sortindex`, at the place of indexing, and
- `\pasteupindex`, for the pasteup of the index.

```
Types of IR
 0  ^{return}
 1  ^|verbatim|
 2  ^|\controlsequence|
 3  ^\<syntactic quantity>
Accents ^{\'el\`eve!},
font changing ^{\bf bold}
and spaces ^{control\ symbol}
Control sequences
  ^{\TeX, and \AmSTeX}
  ^{Lamport and \LaTeX}
brackets ^{\tt< \rm and \tt>}
\newpage ^{return}
\newpage ^{return}%on purpose
\sortindex\pasteupindex\bye
```

`<` and `>` 1.
bold 1.
control symbol 1.
`\controlsequence` 1.
élève! 1.
Lamport and LATEX 1.
TEX, and AMS-TEX 1.
return 1–3.
⟨syntactic quantity⟩ 1.
`verbatim` 1.

The representation of page numbers as a range comes out automatically.

7.1 What makes a good index? Of course this is a million-dollar question. Let us concentrate on the number of entries and on the number of page numbers per entry. Which of the two extremes sketched below is the better one in your opinion? One with many entries pointing to issues spread throughout the book—like *The TeXbook* ;-))), and pushing the limits just for the imagination, an index with pointers to related work on the internet, accessible by just clicking the mouse—or one with few page numbers per entry[56]?

Markup of Index Reminders

IR-s are at the heart of the process. Knuth distinguished 4 types to facilitate the outside processing. I'll adopt his IRs syntax and types.

Syntax Knuth's IRs obey the following syntax.

$$\langle \textit{word(s)}\rangle_{\sqcup}\texttt{!}\langle \textit{digit}\rangle_{\sqcup}\langle \textit{page number}\rangle\texttt{.}$$

The digits 0, 1, 2, or 3 denote the types: words, verbatim words, control sequence, and syntactic quantity. A user does not have to bother about the digits nor about the page numbers. Knuth has adopted the accompanying conventions for the word(s) of IRs.[57]

<table>
<tr><th>Mark up</th><th>Typeset in copy*</th><th>IR</th></tr>
<tr><td><code>^{...}</code></td><td>...</td><td><code>... !0</code> ⟨page no⟩.</td></tr>
<tr><td><code>^|...|</code></td><td><code>|...|</code></td><td><code>... !1</code> ⟨page no⟩.</td></tr>
<tr><td><code>^|\...|</code></td><td><code>|\...|</code></td><td><code>... !2</code> ⟨page no⟩.</td></tr>
<tr><td><code>^\</code>⟨...⟩</td><td>⟨...⟩**</td><td><code>... !3</code> ⟨page no⟩.</td></tr>
</table>

* `|...|` denotes manmac's, TUGboat's,...verbatim
** in `\rm`

For the user the word(s) is (are) important. The markup allowed for the IRs and the result in the copy are given in the accompanying table.

Markup The markup for IRs is near to natural. Precede the entry by a circumflex, or a double one in case of a silent[58] index entry.

Example *(IR markup)*

```
^{\'el\`eve!}^|verbatim text|^|\controlsequence|^\<a metalinguistic variable>
^^\<a metalinguistic variable>  %for silent ones, double the ^
{\sl^{ligatures}} |'$|^|\,||$''|%from the TeX book script
^^{markup commands, see control sequences}
^{Lamport and \LaTeX}           %text and control sequences with sort keys
```

Spaces are difficult as always. In the IR they separate parts of the IR and are used in the word part.

- Just typing a space has as an effect that it will be neglected during sorting
- The markup '\␣', a control space, will yield a space subject to sorting, according to the ordering table
- `\space` as markup will be neglected during sorting. This token is default member of the set of control sequences to be ignored. It will be set in the index as \␣.

7.2 What to do when part of a title should reappear in the index?

---

[56] Courtesy Erik Frambach.

[57] See *The TeXbook* 424, for the IR types, and what is typeset in the result. In `\vref` the markup is inserted as replacement text of `\next`. What is set in the index is governed by the macros which are included after `\begindoublecolumns` in the TeXbook script.

[58] Silent IRs mean that these will appear only in the index, not on the page.

Example *(Spaces)*

```
^{\space}%an ignored cs
^{a\ a}  %control space
^{aa}
^{a\ b}
^{a \TeX}
^{a\ \bf a}
^{\TeX book}
^{xyz beta}%space neglected in
           %sorting
^{xyza}
^|\space|
\sortindex\pasteupindex\bye
```

Sorted result in file index.srt

```
\space {} !0 1.
a\ \bf a{} !0 1.
a\ a{} !0 1.
a\ b{} !0 1.
aa{} !0 1.
a \TeX {} !0 1.
space{} !2 1.
\TeX book{} !0 1.
xyza{} !0 1.
xyz beta{} !0 1.
```

Explanation. `\space` belongs to the set of control sequences to be ignored, ICSs for short. This means that it is skipped with respect to sorting, except when it occurs as the last token of the word part. In that case they are ordered as a space, i.e., according to the lowest value. This explains the position of '`\space`.'

'`\TeX`,' and '`\TeX` book,' are subject to the default sorting keys.

'xyza' precedes 'xyz beta,' because the space is silent. When word ordering is preferred a `\␣`, a control space, must be included.

Special tokens

Tokens are either neglected or replaced by another sequence while sorting. blue.tex provides two sets of tokens to be ignored while sorting: `\conseqs` and `\consyms`.[59] Replacing a control sequence by another sequence is called associating a sorting key to the control sequence.

Active symbols can't be part of the IR, for the moment.

Tokens to be ignored In practice I needed things like `\tt` as part of the IR, which must be neglected while sorting.[60] I decided to ignore those tokens while sorting and to include the tokens in the final index.elm as such. Default blue.tex knows about the following sets of tokens to be ignored.

```
\conseqs{\c\space\bf\it\rm\tt\sub\relax}
\consyms{\`\'\"\^\~}
```

Sorting keys In order to extend a set, use the macro `\add`.

Example *(Use of sorting keys)*

Default blue.tex provides the following sorting keys.

```
\srtkeypairs{\AmSTeX{amstex}
             \LAMSTeX{lamstex}
             \LaTeX{latex}
             \TeX{tex}
             \PS{PostScript}}
```

Suppose that we have `\fourtex` and that we like this to be sorted as '4tex.' This can be done by extending the set of `\srtkeypairs`, as follows.

---

59 There are two sets because of the handling of the space after the token in the result.

60 The reason is that `<, and >` are used, and printed wrongly.

```
\add\fourtex{4tex}to\srtkeypairs
Copy with ^{IR \fourtex}
^{IR 1}
^{IR 5}
^{IR a}
%
\sortindex   %with 4tex for \fourtex
\pasteupindex%Set ‘IR \fourtex{}
             %<pagenumbers>’
\bye
```

then the file index.srt will contain the IRs

IR 1 !0 ⟨*pageno*⟩
IR \fourtex{} !0 ⟨*pageno*⟩
IR 5 !0 ⟨*pageno*⟩
IR a !0 ⟨*pageno*⟩

with `\fourtex` sorted on 4tex.

7.3 What to do when ‘to’ is part of the sorting key?

Ordering

A fundamental issue with indexes is the ordering. The ASCII table is not suited because lowercase and uppercase letters differ by 32. I decided to rank these as equal, more precisely to assign the lowercase ASCII values to both. I prefer from the accompanying table the 1^{st} column to the 2^{nd} one.

Moreover, accented letters are not part of ASCII. How should we order for example e, é, è, ê, ë? I decided to rank accented letters equal to those without an accent, because I prefer from the accompanying table the 3^{rd} column to the 4^{th} one.

I know that non-letters precede letters, but what about their relative ordering? I decided to stay as close as possible to the ASCII ordering.

Then there is the problem of digits. In IRs they come as part of the word(s) and as page numbers. For the latter I used the numerical ordering. For the former I used the alphabetical ordering.[61]

Furthermore, a user can select the so-called word ordering,[62] by `\␣`, TEXnically a control space, as markup for a space. Personally, I like from the accompanying table the 5^{th} column better than the 6^{th}.

lower vs. upper case		accents vs. unaccented		word ordering	
el	el	el	el	sea lion	seal
Elève	em	élève	em	seal	sea lion
em	Elève	em	élève		

Typesetting the index

The specifications for typesetting a blue.tex index are

- represent the four IR types the same as in the TEXbook
- set in two-columns, balanced, possibly preceded by one-column copy
- set subsidiary entries analogous to the TEXbook
- indent continuation lines by 2em
- indent subsidiary entries by 1em

Users can edit index.elm—read: add markup—and provide the necessary macros in for example `\preindex`. In short follow Knuth. To please Frans Goddijn I introduced the tag `\numberstyle`, by default equal to `\oldstyle`.

[61] I could have applied a look ahead mechanism and use numerical ordering throughout. Maybe another time.
[62] This means that a space precedes all letters. A space as such is neglected in the ordering.

Customization

A user might wish to interfere in places

- to include other tokens to be ignored while sorting
- to supply an ordering of his/her own
- to enrich the sorted and compressed file index.elm.

Adding tokens What are reasonable requirements to impose upon the handling of markup control sequences (cs for short)? In my opinion

- the cs must be defined
- `\makexref` writes the cs unexpanded
- ordering? unknown, and therefore must be supplied
- `\setupnxtokens` guards that the cs-s are written to index.srt and index.elm.

As a consequence I decided to neglect the 'in between' control sequences while sorting. For those who favour a one-pass job, I have provided the following, though.[63]

The extension of a set of tokens can be done via

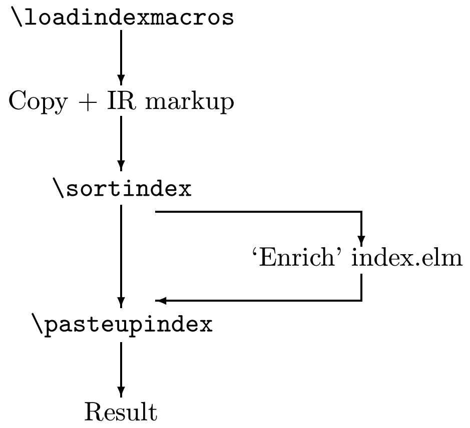
```
\add\hfil to\conseqs  or  \add\`to\consyms  or  \add\hfil{hfil}to\srtkeypairs
%with auxiliary \def\add#1to#2{...}
```

Each element from `\conseqs` is redefined in such a way that the control sequence token is written to the file with a space appended.[64]

Modifying ordering A general way is to 'copy' the ordering table and to modify it.[65]

And what about a macro to add to the table? This can be done easily, and superficially looks convenient for an innocent user. At the moment I don't trust the macros to be worthwhile for an innocent user, unless a very modest index has to be made. And this completes the circle: different ordering is not wanted, I guess.

The process and files involved Like in manmac, blue.tex stores the raw IRs in the file index. The file index[66] is read and stored in an array for internal sorting. After sorting, the number of entries is reduced,[67] and the result is written to the file index.srt. Then, index.srt is transformed into the file index.elm.[68] The result is typeset via `\pasteupindex`. Schematically it comes down to the following.



[63] It is simpler to add those control sequences to index.elm.

[64] `\noexpand` is used instead of `\string`.

[65] My `\fifo` is just a shortcut, which also prevents typos in assigning the ASCII values. For `\fifo`, see my 'FIFO and LIFO sing the BLUes.'

[66] Default index is the value of the toks variable `\irfile`, which is used in `\sortindex`.

[67] Those which differ by page number are collected in one entry.

[68] Default index.elm is the value of the toks variable `\indexfile`, which is used in `\pasteupindex`. The transformation abandons the IR syntax. The part which specifies the kind of IR is deleted and the word part marked up accordingly.

`\loadindexmacros` loads the index and sorting macros, and performs initializations. It is safeguarded against double loading.[69]

Enriching the index This use is necessary when for example

- control sequences have to be typeset
- special symbols are needed, or
- cross-references within the index are required.

The best way is to start from the index.elm file.

Typesetting the enriched file When the default name is used—index.elm—just say `\pasteupindex`. For another file name assign this name to the toks variable `\indexfile`, prior to the invocation of `\pasteupindex`.

[69] I introduced this because I start each chapter with `\loadindexmacros`, independent from whether it is run on its own or as part of the total.

Customization

Adaptation of BLUe's format with respect to the look-and-feel has been discussed, i.e., adapting the page size, the headlines and footlines, and the various headings. Transformation of a BLUe script into a MAPS submission has been touched upon. At the end a quick look has been given into the Pandora box of processing scripts in other languages.

The first thing to do is to personalize the format. Make yourself known. In order to do so look in the file blue.tex for the comment line that starts with `%Personalization`, and adjust what follows.

Customization comes down to adapting TEX's and blue.tex's parameters. The big deal of customization is to adapt BLUe's format for processing scripts in your language. Apart from trivia this is beyond the scope of this guide.

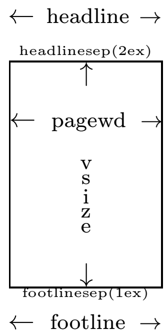
Changing the look-and-feel

Much of the look-and-feel is determined by the page size, the headline and footline, the section beginnings, and the opening part.

Adapting the page size A page is composed of the page proper block, `\pagewd`×`\vsize`, preceded by the headline and followed by the footline. `\hsize` is the width of a column. `\vsize` is the length of a column with footnotes included.[70] Customization of the page dimensions can be done by adapting `\A4`.[71] `\twocol` has been included below, along with the picture `\bluepagepic`.

Example *(Page layout, and two-column settings)*

```
\def\twocol{\maxcols=2 \xcol=1
    \onecolfalse\A4
    \hyphenchar\tentt='055
    \hyphenchar\ninett='055
    \hyphenchar\eighttt='055
    \tolerance500
    \hbadness=499
    \hfuzz=5pt
}
```



Names are inherited from *TUGboat*'s output routine. `\xcol` denotes the number of the actual column. The value of the `\hyphenchar` allows hyphenation for words in those fonts. The tolerance and badnesses are set to circumvent superfluous messages, at the expense of less optimum results.[72]

`\hoffset` and `\voffset`, the offsets of the left upper corner of the page to the boundary of the paper, are both 1 cm less than plain's default of 1 in.

`\A4` reads as follows.

---

[70] The length of the page can actually become a little shorter or longer because of the depth of the last box.
[71] Customization of the page composition is beyond the scope of this guide and requires knowledge of the output routine. `\A5`, or other sizes, can be added.
[72] For the pragmatic approach of getting rid of overfull boxes, I'm much indepted to Phil Taylor.

```
\def\A#1{\csname A#1\endcsname}
\ea\def\A4{\ifonecol\hsize16cm\pagewd\hsize
  \else\hsize9cm\intercolwd.5cm
       \pagewd\hsize\advance\pagewd\hsize\advance\pagewd\intercolwd
  \fi\colwd\hsize \vsize25cm}
```

The header and footer are generated from the token variables `\headline`, and `\footline`. Default they read as follows.

Example *(Default headline and footline code)*

```
\headline={\hbox to\pagewd{\it\the\issue\hfill\the\title}}
%
\footline={\hbox to\pagewd{\fiverm\rlap{Draft \today}\hfill
   \dash\the\pagenumber\dash\hfill\llap{\copyright cgl}}}
```

An extra dimension `\pagewd` is needed in order to have a quantity independent of the number of columns.

8.1 What are the `\headline` and `\footline` for `\report`?

8.2 How can we use the `\headline` for transparencies?

8.3 In a book the footline contains usually the page number away from the back, that is, right on right-hand pages and left on left-hand pages. How can we achieve this?

Title part The markup of the beginning—title part—and the markup of the end—closing part—of a regular BLUe script are implemented as follows.

Example *(Beginscript and endscript code)*

```
\def\beginscript{\lastscript\the\everyscript\begingroup
   \the\thisscript
   \hrule\kern2ex\noindent
   {\titlefont\the\title{\subtitlefont\the\subtitle}}
   \medskip\the\author
   \medskip\pasteupkeywords
   \medskip\pasteupabstract
   \medbreak\pasteupcontents\smallbreak}
%
\def\endscript{\makesignature\xcol=\maxcols \vfil\eject\endgroup
   \tracingstats1\stop\thisscript{}}
%with at the lower level
\def\beginkeywords{\setbox\keywordsbox\vbox\bgroup\subhead{\sevenbf
   \the\keywordsname: }\small}
\def\endkeywords{\egroup}
%
\def\beginabstract{\setbox\abstractbox\vbox\bgroup\centerline{\sevenbf
   \the\abstractname}\beginquote\small}
\def\endabstract{\endquote\egroup}
%
\def\begincontents{\setbox\contentsbox\vbox\bgroup\small\noindent
   \the\contentsname\par\obeylines}
\def\endcontents{\egroup}
%
\def\pasteupkeywords{\box\keywordsbox}
\def\pasteupabstract{\box\abstractbox}
\def\pasteupcontents{\unvbox\contentsbox}
```

Customization can be done by changing the above. I have omitted the macros, which allow the use `\`⟨*tag*⟩`{...}`, the short variant. Whatever your change of `\begin`⟨*tag*⟩ or `\end`⟨*tag*⟩, the short variant `\`⟨*tag*⟩`{...}` remains invariant.[73]

8.4 How can we suppress the author information at the end of the document?

8.5 Enumerate the token variables which contain language-dependent names.

8.6 How can we obtain Roman numerals as page numbers?

8.7 How can we suppress the headline on the first page?

8.8 How come that more than one script can be formatted despite `\stop` in `\endscript`?

For parameters which govern special constructs, such as the shape of paragraphs, items, tables, or ... , see the chapters on the issue.

Section beginnings `\head` can be customized by adaptation of the parameters `\prehead` and `\posthead`, to position the head within context. `\ignorewhitespace` does what its name suggests.[74] The font used is parameterized in `\headfont`.

Example *(Head code)*

```
\prehead{\vskip0pt plus5ex\penalty-250\vskip0pt plus1ex
   \bigskip\noindent}
\posthead{\medskip\nobreak\noindent\ignorewhitespace}
\def\beginhead{\the\prehead\bgroup\headfont}
\def\endhead{\egroup\the\posthead}
```

`\subhead` and `\subsubhead` are constructed similarly.

Raster layout and TEX As Nico Temme communicated to me publishers require sometimes fixed white space between headings and the surrounding text. How to achieve this? Naively this looks simple, just provide `\prehead` and `\posthead` with the fixed skips. The problem is, however, that TEX likes some freeway to shrink or stretch a little.

BLUe script into MAPS submission

In principle this is not too laborious, because only a handful of outer-level control sequences need adaptation.[75]

MAPS.sty is LATEX biased. To give you an impression of the amount of work a template for a MAPS submission and a transformation table is given below.[76] Because BLUe's format is a personalized format you have to add `\author{...}` and the like.

Globals deserve special attention, because they might interfere with other submissions.

The title block is filled in or simply reused from the template. No problem at all. A table of contents can be moved to the back part.

I fell back upon LATEX's bibliography handling.[77] My use of so-called 'run-in' heads requires special attention.[78] Important is that blue.tex's verbatim suite, tables and pictures can be reused as such, similar to plain's math constructs. For other things like my way of citation—⟨*author*⟩, ⟨*year*⟩—no adaptation is needed.

---

73 Consult the paper 'Paradigms: Two-part macros' for the explanation.

74 It gobbles white space after `\beginhead` up to the head title.

75 In general transformation is very complex and confusing, for example if we also consider automatic generation of an index. It is next to impossible to do this automatically. When submissions to (various) journals are in sight, keep the markup as simple as possible. And why not stick to simplicity as a rule?

76 I don't use times fonts, don't use Babel (no dutch.sty, nor english.sty), no fancyheadings are involved, don't need Mittelbach's multicolumn, nor Schöpf's verbatim facilities, and the like. Until today MAPS has used LATEX2.09. Minimal markup tags have been neglected.

77 Nowadays, I tend to omit the references in a MAPS submission, and refer to my database lit.dat.

78 The (sub)subhead title is part of the first sentence of the section.

Example *(BLUe script into MAPS submission)*

```
%Template for MAPS submissions
\documentstyle[twocolumn]{maps}
\begin{document}
\title{...}
\subtitle{...}
\author{...}
\date{}
\maketitle

\begin{abstract}...\end{abstract}
\begin{keywords}...\end{keywords}
%Contents table
%Copy proper
\head{...}%et cetera
%back matter, LaTeX's
\begin{thebibliography}{abcde}
...
\end{thebibliography}
\end{document}
```

BLUe script		MAPS.sty script
`\begin` `\end` abstract	⟶	`\begin{abstract}` `\end{abstract}`
`\begin` `\end` keywords	⟶	`\begin{keywords}` `\end{keywords}`
`\beginscript`	⟶	`\begin{bijlage}`
`\head`	⟶	`\let\head\section`
`\bitem` ... `\bitem`	⟶	`\begin{itemize}` `\item...` `\item...` `\end{itemize}`
`\begin` `\end` quote	⟶	`\begin{quote}` `\end{quote}`
`\ftn`	⟶	`\footnote`
`\begin` ⟨*escape char*⟩`end` verbatim	⟶	`\begin{verbatim}` `\end{verbatim}`
`\endscript`	⟶	`\end{bijlage}` `\end{document}`

Other languages

The Pandora box of customization is to tailor BLUe's format towards processing scripts in your language. Knuth adapted TEX in 1989, and provided `\language` to handle hyphenation per language, and smarter ligatures, if I'm not mistaken. The trivia for adaptation are changing the names for document parts, like 'Abstract, Keywords,' and similar things. In BLUe's format the latter are parameterized in token variables.

More complicated is creating an index on the fly in your language via BLUe's format.[79]

An interesting application of formatting simultaneously two (or more) languages is: two-column, with each column devoted to a language. That is beyond the scope of this guide.

[79] For the time being this is open. The language-dependent ordering of letters has been parameterized in an ordering table though.

Afterthoughts

This chapter relates BLUe's format to existing formats and styles, especially $\mathcal{AMS}$-(La)TeX, (La)TeX, LAmSTeX, and the TUGboat styles. At the end some statistics about processing PWT has been included.

Why this work? Is LaTeX not sufficient? LaTeX was not necessary, the TeXbook contained it all. :-))). This advantage turned out to be a severe disadvantage. TeX was simply too much, too unusual and ahead of its time. The bad news is, alas, that TeX is very error-prone, in the hands of ordinary people.[80] One has to be very precise, has to oversee all the consequences, and has to be utmost, utmost consequent.

Perhaps because of the latter Lamport dared to do less, thought that simplifying is the way out. To quote Dijkstra

'Abstraction is our only mental tool to
master complexity'

He provided a system tuned for the end user, along with a user's guide. The latter is the big deal, for general acceptance.

If only there had been a user's guide for manmac,
the (LA)TeX world would have looked much different
from what it is today.

Below I'll very roughly characterize other formats, which enjoy a general acceptance, in relation to BLUe's format. The AMS and TUG styles, and the work of Spivak, have been surveyed—and commented in depth—in separate articles.

$\mathcal{AMS}$-(La)TeX

These formats are not personal but provided and maintained by the American Mathematical Society. It is a rich and wealthy source. My comments date back to 1992, when I studied the formats in detail. Their user and installation guides are very good. The coding is clever, very clever, but tends to be monolithic. It is not easy to borrow parts from it. The formatting of references in blue.tex has been inspired upon $\mathcal{AMS}$-TeX's preprint style.

LA$\mathcal{MS}$-TeX

Michael Spivak did a great job. His macros for tables and commutative diagrams are impressive, very impressive. The way he recast the LaTeX functionality looks very promising. It is only a pity that his code does not enjoy general acceptance, as far as I know. It is even sadder that his codings have not been studied nor borrowed from.

[80] Don't misunderstand me, the software itself is as free of errors as bug free can be. No, it has all to do with people who partly understand the system, but need macros. Many a released macro—the more so for collections of macros—is immediately followed by an improved version, and so on. The last observation is due to George Greenwade, and I agree with him.

LaTEX

Indeed, LATEX is not sufficient, although much of the functionality of BLUe's format was already provided by LATEX. The setup of BLUe's format is much different though. LATEX does not suit my purposes. Its Achilles heel is customization. Some features of BLUe's format are its open eye for change, its open-endedness, next to its databases, that is what I need. I understand blue.tex and feel happy with it, :-))), probably for the rest of my life. The great—not to be underestimated—virtues of LATEX are

separate style files

user's guide

general acceptance

BLUe's format is stable, compatible with plain TEX, well-organized, and reasonably compact. I have tried to combine the best of both worlds.

Stay with Knuth with respect to his unsurpassed markup, and

borrow Lamport's awareness of the necessity of a user's guide.

If you ponder about a manmac user's guide, it is mandatory to pay attention to developments since.[81] That is how BLUe's format emerged. A tribute to manmac and some more, that is the royal road I passed by. Reality shows that manmac.tex has been neglected by the (LA)TEX community. Given the reputation of Don Knuth, and the wide-spread use of LATEX, I'm realistic about the acceptance of BLUe's format.[82] Nevertheless, I hope that BLUe's format will find its niche within the spectrum of tools for computer-assisted production and consumption of information. IMHO, with all respect, it deserves it. :-))).

In the past I used LATEX but was unhappy with the layout, and therefore added modifications, like many of us. The results did not satisfy me, because I never got the feeling that I understood the coding. Even worse, I never felt happy with the coding of LATEX.

Since then, I have become more and more enthusiastic about Knuth's markup, especially about the subtleness of plain and manmac. Understanding his macros is a real challenge, but once mastered you feel like in heaven. It gave me such a pleasure. The beneficial spin-off of this thorough study was that the knowledge gained contributed much to the compactness of BLUe's format.

At the EuroTEX 94 the new, long-awaited LATEX2ϵ went public. During the conference I was asking myself over and over the same questions, to uncover the reasons why I don't believe that the approach taken is the right one, especially for self-publishing authors. My septicisms come down to the following.

- LATEX is essentially quasi-static in contrast with plain, which is frozen and hence stable. I know that dynamics is the nature of progress, but only a small amount needs change. The lack of the awareness to separate the quasi-static part from the stable part, is what I don't like.
- The macros are essentially one-part macros with flexibility via optional and ordinary parameters. The mechanism of parsing optional arguments, is a non-TEX approach, at the expense of cumbersome and monolithic coding. Two-part macros should be at the basis, with options via token variables. As far as my experience goes this will yield simpler coding, with the benefits of easier maintenance.
- LATEX is verbose and far from minimal markup
- LATEX is not easy to customize
- LATEX's macros are incompatible with plain, even the use of `\def` is not possible[83]
- LATEX turns out to be a cuckoo's egg

[81] However, I must confess that I'm not completely up to date with PostScript, Adobe's Photoshop, nor Acrobat.

[82] Thinking aloud: I would be happy if 50% of NTG's members finds something of interest in BLUe's format, be it the system, a nice macro, or just an eye-opener with respect to markup or computer-assisted typesetting in general. From CyrTUG I expect that even 75% of their members can take advantage of the system. And what about 10% of the *unorganized* TEX users, ;-)))?

[83] I know of the annoyance of silently redefined macros, but nevertheless.

- LATEX tends to represent the establishment, not what individuals really need, IMHO, with all respect. The imposing character is believed to be beneficial for publishers, to ascertain that authors don't use markup different from the standard style. Much truth in there, but the world is changing. I favour simple tools, which I understand to a high degree, and which are cooperative.

I only hope that the LATEX developers can take advantage of my views.

TUGboat styles

I learned a lot from those. The functionality of the OTR has been taken over in BLUe's format. It is just a pity that a preprint style is lacking. I also consider the approach of the separate streams TEX next to LATEX, cumbersome to maintain. At the lower level separate collections are in use next to each other. For example the verbatim macros for TEX differ from those for LATEX. A common picture environment for both streams is not there either. Similar to the AMS formats the coding is monolithic. It is not easy to borrow parts from it.

Not a common TEX format with a LATEX user interface on top.

I have used them for quite a while, but once I looked into the code, I decided to go over to the manmac line of thought, back to the roots, so to say, with all respect.

The future

More important, and thinking positively, is to day-dream about the future and especially to ponder about the new media to assist information production and consumption.

It is a little peculiar that I sympathize so much with the new media, the newer developments, which will make it possible for the

masses to access information easily,

the more so because this file is much like a traditional book. It does not combine different media like text, sound, colours or graphics in general. Not even does it allow you to click through the file in a hyperlink sense. I'm not ashamed of that. There are so many problems to be solved, before we come even close to coping better with information, I mean taking real advantage of the information.

The buzzword which goes along with this is the 'electronic highway,' having rapid and cheap access to information stored digitally. Perhaps the words 'database publishing,' and 'multi-media' appeal even more. Whatever the value of these approaches and promises, let us down-to-earth ponder about electronic equivalents of the old goodies, such as a table of contents, an index, and cross-references.

A Table of Contents is getting more and more important. When tables of contents, ToCs for short, are generated automatically, which is nice, they can't be searched for in a script. A surprising consequence is to store a ToC also explicitly, and perhaps separately from the work, because people first like to browse through a table of contents or read.me file. For electronic use the page numbers attached are not that important. How to access parts is more important. Therefore the ToC should contain hyperlinks to the real information.

An Index is also no longer tied up with one book. Databases can be searched on keywords, or more selectively browsed via queries, to spot what one is looking for, whatever the context. When we have spotted what we are looking for, another click would be all that is needed to get into it.[84]

Cross-referencing is really outdated. Electronically, we are not limited by the traditional sequential order of numbered pages. Relational links are there to bring you immediately to the material required, be it to more details or to just the opposite, a survey, with fewer details, or, ... another representation, like spoken in a different language.

---

[84] Let us forget, for the moment, the reality of the morning-after invoice.

The bad news is that none of those goodies mentioned are part of this work. Just an afterthought, some wishful thinking, pondering about future work.

Frank Poppe communicated that he would be happy with the TEXbook script with hyperlinks. That he could easily click through the related material spread over the file. Indeed that would be nice.

Statistics about processing PWT

The booklet contains 125 odd pages and some 400 index entries, all processed in one pass. The sorting of the index did take an hour on my Mac Classic II, and 10K words of memory. For re-runs without page changes one can do without `\sortindex` because `\pasteupindex` reads from index.elm.

```
967 strings out of 5259
4438 string characters out of 31539
63053 words of memory out of 65536
2531 multiletter control sequences out of 10000
18 hyphenation exceptions out of 307
22i,21n,16p,216b,1015s stack positions out of 200i,60n,60p,5000b,2000s
```

A: Answers to the exercises

1.1 To override the defaults assign values to the token variables `\author`, `\address`, `\netaddress`, and `\issue`.

1.2 Provide `\everyscript{\notlastscript}` and precede the last script by `\thisscript { \lastscript }`. This holds for scripts in sequence. For nested scripts see this script file for the examples of a letter and a few transparencies run within this script.

2.1 `\leftskip`0pt plus1fil, `\rightskip`0pt plus−1fil, and `\parfillskip`0pt plus2fil. End the paragraph with `\hskip0pt plus1fil`.

2.2 Jan Kybic reported about the several ways he received as answers to this problem on comp.text.tex. Below is the solution due to Petr Olsak. Beware it is limited. The general idea in the solutions collected by Kybic is to take apart the lines set in a vbox either starting at the end via `\lastbox` or starting at the begin via `\vsplit`, collect these lines extended by their line numbers, and reinsert these later.

```
%Due to Petr Olsak
%Disclaimer: Macros for plain prose line
%numbering only for ordinary paragraphs
%without display mathematics or other
%specialities.
%(Note added: no \footnote, \centerline.)
\newcount\linenum
\newcount\worknum
\newbox\allparagraph
\def\beginnum{\begingroup\linenum0
   \def\par{\ifhmode\completepar\fi}%
   \setbox\allparagraph\vbox\bgroup}
%
\def\endnum{\par\egroup\endgroup}
%
\def\completepar{\endgraf \global
   \advance\linenum\prevgraf
   \worknum\linenum \setbox0\null
   \loop\unskip\unpenalty
     \setbox2\lastbox
     \ifhbox2 \global\setbox0\hbox
      {\llap{$\scriptstyle\the\worknum.$
             }\box2\penalty0\unhbox0}%
      \advance\worknum-1
   \repeat\egroup\noindent
   \unhbox0\unpenalty\endgraf
   \setbox\allparagraph\vbox\bgroup}
```

Another application is mentioned by Salomon in his ‘Insights and Hindsights’ to number the lines on each page. This is handy for proofers to indicate more precisely the commented parts. See his OTR articles as published in *TUGboat*—or his courseware—for the modified OTR. The numbering of verbatim lines has been treated separately.

2.4 Use `{\par}` instead of `\par`. See their article ‘From `\indent` to `\par`,’ MAPS94.2, among other things. The point is there are changed, but—there is always a but—on encountering a `\par` the paragraph shape parameters are reset *locally*, within the group so to say. After the group closing curly brace the old values pop up again, which yields the effect.

2.5 No, it is not.

2.6 In `\manmacauthor` `\rm` is switched on for the special paragraph of the author line. The next quotation, if there is one, takes `\sl` again, automatically. Note that the `\noindent` in `\manmacauthor` is mandatory, because it starts the new paragraph and therefore `\rm` will be in effect. It will come after the automatically inserted `\sl`.

2.7 `\begincenter...\endcenter` does. `\begincenterblock...\endcenterblock` does not, because it is a `\vbox`.

2.8 Enclosing the construct with a `\vbox` in math display does not work, which is the usual way to center a `\halign`. This comes from the fact that the construct is recognized as an alignment display. A solution is

```
{\def\enddisplay{\crcr\egroup\egroup$$}
  \begindisplay\vbox\bgroup
  ⟨displaymaterial⟩
  \enddisplay}
```

Perhaps it is simpler for those cases to fall back on the following usual markup

```
$$\vbox{\halign{⟨template⟩
  ⟨displaymaterial⟩
  }}$$
```

2.9 The set of reserved words has to be customized, next to the handling of special symbols.

2.10 The naive way introduces an unwanted space. We can get rid of a space after a control sequence via the use of a control symbol with category code 9 (to be ignored) after those control sequences.

2.11 This is included in BLUe's format and reads as follows.

```
\def\boxlines{\obeyspaces\fifol}
%\gdef\fifol#1^^M{\ifx\lofif#1\lofif\fi
%  \processl{#1}\fifol}
\def\begincenterverbatim{$$%
   \def\processl##1{\hbox{##1}}%
   \def\lofif##1\fifol{\fi\egroup
     \egroup$$\thisverbatim{}}
   \let\endcenterverbatim\lofif
   \bgroup\tt\setupcopy
   \let\!=!\catcode`\!=0
   \the\everyverbatim\the\thisverbatim
   \vbox\bgroup\boxlines}
```

2.12 It looks although all depends on the values of `\preverbatim` and `\postverbatim`. The defaults are `\medskip` and `\smallbreak`. The former does not insert a `\par` while the latter does. As is well-known a `\par` terminates a paragraph. However, the subtleness is that `\begingroup` and `\endgroup` enclose the verbatim which, *The TEXbook* 279 'don't change the mode.' On reading I did not understand the consequences of the quoted sentence. A small experiment taught me that the paragraph parameters are not reset. Therefore the `\beginverbatim` ⟨*escapechar*⟩`endverbatim` just *interrupt* the paragraph, similar to the math display.

2.13 This is not implemented in blue.tex's OTR. It is rather difficult to combine that with flexibility with the number of columns. If needed occasionally use `\vadjust`.

2.14 Copy Knuth's `\tenpoint` and adjust for the size.

```
\def\tenpoint{\def\rm{\fam0\twelvenrm}%
  \textfont0=\twelverm\scriptfont0=\ninerm
             \scriptscriptfont0=\sevenrm
  \textfont1=\twelvi \scriptfont1=\ninei
              \scriptscriptfont1=\seveni
  \textfont2=\twelvesy\scriptfont2=\ninesy
             \scriptscriptfont2=\sevensy
  \textfont3=\twelveex\scriptfont3=\tenex
              \scriptscriptfont3=\tenex
```

```
  \def\it{\fam\itfam\twelveit}\def\oldstyle
  {\fam1 \twelvei}\textfont\itfam=\twelveit
  \def\sl{\fam\slfam\twelvesl}%
  \textfont\slfam=\twelvesl
  \def\bf{\fam\bffam\twelvebf}%
  \textfont\bffam=\twelvebf
   \scriptfont\bffam=\sevenbf
   \scriptscriptfont\bffam=\fivebf
  \def\tt{\fam\ttfam\twelvett}%
  \textfont\ttfam=\twelvett
  \tt \ttglue=.5em plus.25em minus.15em
  \normalbaselineskip=14pt minus1pt
  \def\MF{{\manual META}\-{\manual FONT}}%fixed
  \let\sc=\tenrm
  \let\big=\fourteenbig
  \setbox\strutbox=\hbox{\vrule
    height10pt depth4pt width\z@}%
  \normalbaselines\rm}
\def\fourteenbig#1{{\hbox{$\left#1\vbox
   to12pt{}\right.\n@space$}}}
```

Make sure all the used fonts are available, otherwise provide scaled versions. See the script for `\Large`.

2.15 Once you realize that `\bf` and `\it` are shortcuts to the fonts `\tenbf` and `\tenit`, actually to the family, the answer prompts itself: the last font, that is italics respectively bold face roman. One of the features of NFSS is that you will get bold italics in both cases, that is these tags commute, or am I mistaken?

3.1 This is not possible in a one-pass job. Therefore a note is printed in the margin while proofing. The correct number has to be filled in ultimately.

3.2 Provide `\advancefalse`, and `\postnum{a)}`.

3.3 Use `\eqalign` as follows.

```
$$\unitlength1ex
  \eqalign{
   \icmat44\kern\unitlength\iclllt44={}&
      \iclllt44\icuh413&
      \qquad \hbox{AL${}={}$LH}\cr
   \icmat63={}&
      \icmat63\kern\unitlength\icurt63&
      \qquad \hbox{\phantom{A}A${}={}$QR}}
$$
```

3.4 The empty formula {} is used to coerce the + to behave as a dyadic operator, and the = to behave similarly. In other words to yield the correct spacing.

3.5 It is in the wasy font. A poor man's version is an open problem as yet, because the lower 0 is neither 5pt, nor 6pt, and therefore the symbol can't be built from % and an appropriate sized 0. A very, very poor man's version reads `\%\lower.2ex\hbox{\fiverm 0}`, in fact unacceptible. Building it from `\frac`, `\frac0/{00}`, is different from %.

3.6 Automatically! *The TEXbook* 144: ' A displayed sum usually occurs with 'limits,' i.e., with subformulas that are to appear above and below it. ... According to the normal conventions of mathematical typesetting, TEX will change this to '$\sum_{k=1}^{\infty}\frac{1}{2^k}=1$' (i.e., without limits) if it occurs in text style rather than in displaystyle.' Explicit control is possible via the control sequences `\limits`, respectively `\nolimits`.

3.7 To set the curly brace on the baseline.

3.8 It looks like it. Used within the picture environment I stumbled on lack of scaling invariance for the latter case?!? As far as I see it now the `\cdot` yields nicer result anyway. Because of the explicit `\Big` the above markup is not scaling invariant.

3.9 Give the summation symbol depth 0.

3.10 It could have been anything representative for the lower part. With the k and the subscript in the `\phantom` we are sure that the vertical positioning will be OK.

3.11 Define `\lft` and `\rgt` equal to $ with `\hfil` added left, respectively right, for centering.

3.12 Make use of nested tables. Consider the partitioned matrix, a `\ruled\btable`, to be built from blocks with each block a non-ruled matrix.

3.13 This is an open problem. If you have an elegant solution, let me know.

3.14 In pic.dat a trial-and-error solution is included. Not elegant. Improve, enjoy, and let me know!

4.1 The header separator is not influenced by the ruled attributes.

4.2 Just consider all as data.

```
Alternative
\def\data{Horizontal lists    \cs Chapter 14\rs
          Vertical lists\hfill\cs Chapter 15\rs
          Math lists\hfill    \cs Chapter 16}
$$\btable\data$$
```

4.3 The macro and auxiliaries read as follows.

```
\newcount\n
\newcount\rcnt  \newcount\ccnt
\newcount\tentry\newcount\prev
%
\def\pascal#1{\n#1 \def\0{1} \ccnt1
   \loop\ea\xdef\csname\the\ccnt\endcsname{0}
     \ifnum\ccnt<\n \advance\ccnt1
   \repeat\rcnt0 \ccnt0
   \displaylines{\rows}}
%
\def\rows{\global\advance\rcnt1
 \ifnum\rcnt>\n \swor\fi
 \nxtrow\rows}
%
\def\swor#1\rows{\fi}
%
\def\nxtrow{1 \ccnt1 \prev1
 \loop\ifnum\ccnt<\rcnt
  \tentry\prev
  \prev\csname\the\ccnt\endcsname
  \advance\tentry\prev
  \ea\xdef\csname\the\ccnt\endcsname{\the\tentry}
  \quad\the\tentry
  \advance\ccnt1
 \repeat\cr}
```

4.4 Extend the data by the numbers at the end of each row, and copy the last row also on top.

```
\bdata%
*ABCDEFGH*
8 * * * *8
7* * * * 7
6 * * * *6
5* * * * 5
4 * * * *4
3* * * * 3
2 * * * *2
1* * * * 1
*ABCDEFGH*
\edata
```

4.5 Make the border thicker. For example via `\boxit{\boxit{\sol}}`. Why don't you develop a nice border macro?

4.6 Chess pieces are available in a font. Similar to the letters in the solution of the crosswords we can put in pieces. For example let us assume for simplicity's sake that K denotes the white king then the following yields the requested result.

```
\begincrosswords
\obeyspaces\let =\space
\csize=3ex
\bdata%
8 * * * *
7* * * *
6 * * * *
5* * * *
4 *K* * *
3* * * *
2 * * * *
1* * * *
*ABCDEFGH
\edata
\boxit{\boxit{\sol}}
\endcrosswords
```

4.7 Descriptive 1, visual 2, and for plain TEX formatting 3. It is all a matter of point of view.

4.8 The 'data' can be defined as follows.

```
\def\data{1.\cs\cs\cs\cs\cs\cs\rs
          2.\cs\cs\cs\cs\cs\cs\rs
          et cetera\hidewidth}
```

4.9 Add `\hfil` in the `\rowstblst` via

```
   \def\rowstblst{{\hfil1.}{\hfil2.}{et cetera}}
```

Another solution is the following template

```
   \def\template{\colsepsurround\hfil##\hfil&&\lft##\rgt\cr}
```

4.10 As a parameter I provided the `\hsize` of the table entries. For row spans the `\logms`, logical multispan, has been used, similar as for `\btable`.

```
\def\xsp{\hbox{ }}
\def\vruled{\def\colsep{\vrule}\def\rowsep{}}
\def\hruled{\def\colsep{}\def\rowsep{\hrule}}
\def\ruled{\def\colsep{\vrule}\def\rowsep{\hrule}}
\def\nonruled{\def\colsep{}\def\rowsep{}}
\def\framed{\let\frameit\boxit}
\def\nonframed{\def\frameit##1{##1}}
\def\rs{&\null\rowsepsurround\rowsep\rowsepsurround&}
\def\cs{\lcs\noalign{\colsep}}
\def\lcs{\rowsepsurround\cr}
\def\top{\null\vskip.5em}
\def\bot{\vskip.5em}
\def\ctr{\def\lft{\hfil}}
\def\fll{\def\lft{}}
\def\flr{\def\lft{\hfill}}
%Macro
\def\vtable#1{\frameit{\hbox{\valign{\top\noindent
   \hsize#1\lft##\bot
   &&##&\hsize#1\top\noindent\lft##\bot\cr
  \data}}}}
%Defaults
\nonframed\nonruled\ctr
```

```
\def\rowsepsurround{\vskip.1ex}
%Examples
\def\data{1 \rs  2\rs 3\cs
          4 \rs  5\rs 6\cs
         314\rs271\rs  \lcs}
\vtable{7ex}
\smallskip
\def\data{\logms3{\hsize7ex\null
\vfil\noindent\hfil1\hfil\vfil}\cs
          C\rs G\rs L\cs
         22\rs12\rs1943\lcs}
\ruled\vtable{7ex}
\smallskip
\def\data{\logms3{\hsize15ex\null
\vfil\noindent\xsp This\\
              \xsp looks\\
              \xsp problematik\vfil}\cs
                       O\rs  A\rs G\cs
                      16\rs11\rs1954\lcs}
\framed\ruled\vtable{5ex}
```

4.11 Because of the separation of concerns we only have to concentrate on the data. My approach is to look at a block—here 2∗3—as a ghost to be overprinted. A natural way is to introduce the concept of a ghostrow, which will be invoked by the ghostrow separator `\grs`. So first mark up for the data with the hole in the table and include the `\grs`. Then provide `\ghostrow`. In order to do the latter gracefully, we also need the concept of partial row separator, `\prs`, and its mirror, `\srp`, to handle partial lines.

```
\setbox\block=\hbox{$2*3$}
\def\data{\logms3\cs14\grs
          \logms3\cs24\rs
          31\cs32\cs33\cs34}
\def\btablecaption{Caption}
\def\rowstblst{{row 1}{row 2}{\hfil
  $\Rightarrow$}}
\def\header{\logms4 Header\hfil$\Downarrow$}
\def\footer{Footer}
\thisbtable{\def\ghostrow{\ifx\empty\rowstblst
  \else\omit\lineglue\ea\srp\fi
  \logms3\hfil\vbox to0pt{\vss
  \copy\block\vss}\hfil\prs\lineglue\cr\nxtrs}
}\vcenter{\framed\ruled\btable\data}$$
%
%and the 3*3 block can be obtained via
%
%\setbox\block=\hbox{$3*3$}
\def\data{\logms3\hfil\cs14\grs
\logms3\hfil$3*3$\hfil\cs24\grs
       \logms3\hfil\cs34\rs
           41\cs42\cs43\cs44}
\thisbtable{\def\ghostrow{\ifx\empty\rowstblst
  \else\omit\lineglue\ea\srp\fi
  \logms3\hfil\prs\lineglue\cr
  \ifx\empty\rowstblst\else\ea\nxtrs\fi}}
\def\btablecaption{Caption}
\def\header{\logms4 Header\hfil$\Downarrow$\ }
\def\rowstblst{{row 1}{row 2}{row 3}{\hfil$\Rightarrow$}}
\def\footer{Footer}
$$\vcenter{\ruled\framed\btable\data}$$
```

4.12 A solution reads as follows (Similar approach as to Pittman's multiplication tables.)

```
\newcount\rowno \rowno1
\newcount\columnno \columnno1
\newcount\tnum
\newcount\diff \diff1
\newcount\n
\def\rows{\global\tnum1
     \global\advance\diff1
   \cols
   \ifnum\rowno=\n\lrs\swor\fi
   \rs\rows}
\def\swor#1\rows{\fi}
\def\cols{\the\tnum
   \global\advance\tnum\diff
   \ifnum\columnno=\n\sloc\fi
   \cs\cols}
\def\sloc#1\cols{\fi}
\def\cs{\hss\vertrule \ifnum
   \columnno<\rowno width0pt\fi\relax
         &\global\advance\columnno1 }
\def\rs{\hss\vertrule\relax\cr
     \global\advance\rowno1
     \global\columnno1 }
\def\lrs{\hss\vertrule\relax
     \global\columnno1
     \global\rowno1
     \global\diff1 \cr}
%
\def\gnonoms#1{\n#1 \offinterlineskip
\halign{&\vbox to12pt{\hbox to15pt
           {\hss\strut\oldstyle##}\vss\horrule
            \ifnum\rowno<\columnno
             width0pt\fi\relax}\cr
          \rows}}
\let\vertrule\vrule
\let\horrule\hrule
$$\vtop{\gnonoms2}\qquad
  \vtop{\gnonoms5}\qquad
  \vtop{\gnonoms{10}}$$
\bye
```

4.13 The following redefinitions yield the results.

```
\def\vertrule{\vrule width1ex}
\def\horrule{\hrule height 1ex}
\def\cols{\ifnum\columnno=\n\sloc\fi
   \cs\cols}
$$\vtop{\gnonoms3}\qquad
  \vtop{\gnonoms5}\qquad
  \vtop{\gnonoms7}$$
```

4.14 Similarly, as follows.

```
\font\tenln=line10
\font\tenlnw=linew10
\beginchart{\postdisplaypenalty=0
            \tenln}
   \normalchart
\endchart
\beginchart\tenlnw
   \normalchart
\endchart
```

4.15 There is no much use in providing an optional one. The font is needed and *must* be specified.

5.1 The common-for-pictures macro, `\commonforpic`, is loaded, and invoked. The latter means that you can use for example the macros `\corners`, i.e., the macros which are common for several pictures, and these can be used over and over.

5.2 Quintic spline TEXnique. Open problem as yet.

5.3 blue.tex contains a macro `\contentsdatabase` that writes the list of names of a lit, pic, or address database to the file contentslit, contentspic, or contentsaddress.
Let us assume that we wish the list of names in a toks variable `\namelst`. To facilitate processing of `\namelst` `\lst` has been inserted before each name in the list.

```
\newtoks\namelst
\def\lst#1#2{\namelst\ea{\the\namelst\lst#1}}
\input pic.dat
%and to verify the contents
\baselineskip12pt plus2pt minus1pt
\lineskip1pt plus 1pt minus1pt
\def\lst#1{{\tt\string#1}\endgraf}
\the\namelst
```

5.4 The important parameters for this macro are the coordinates of the upper right corner of the card. Given that point we can draw lines, put text somewhere et cetera. The macro has been included in the 'common-for-pictures' part of pic.dat to be used in pictures.

```
\def\card#1#2#3{%#1,#2 x,y coord.corner; #3 label
   \put(#1,#2){\line(-1,0){1.25}}
   \put(#1,#2){\line(0,-1){ .45}}
   \put(#1,#2){\lower.45\unitlength
     \hbox{\line(-1,0){ .25}}}
  \put(#1,#2){\hbox to0pt{\kern-1.25\unitlength
     \hbox{\line(0,-1){ .25}}\hss}}
  \put(#1,#2){\lower.5\unitlength\hbox
  {\hskip.05\unitlength{\tiny #3}}}
}%end card
```

5.5 The important parameters for this macro are the coordinates of the first point of the right-upper diagonal ellipsis. The macro has been included in the 'common-for-pictures' part of pic.dat to be used in pictures. The basic part from below is borrowed from *The TEXbook* ex18.45.

```
\def\diagdots#1#2{\put(#1,#2){$.^{.^.}$}}
```

5.6 It is not a problem to complete the circle. An open problem is how to specify and implement the parts of the pie-chart, especially each size. If you have found a graceful solution, let me know. Perhaps we should let it go because of PostScript, where it can be done easily.

6.1 Via this empty specification we can typeset what has been supplied in `\thisreferences`. Agreed, a little unusual.

6.2 Recognize that the file consists of triples. BLUe's format provides `\setdatabase` for this purpose. That macros comes down to the following.

```
\input blue.tex \onecol
\prenum{[\thinspace}\postnum{\thinspace]}
\def\lst#1{\nitem}
\input lit.dat
\smallbreak
\bye
```

An older fifo-oriented solution reads as follows.

```
\newcount\bcnt
\def\prt{\medskip\noindent\advance\bcnt1
     \llap{[\number\bcnt]\quad}}
\def\fifo#1#2#3{\ifx#1\ofif\ofif\fi
\prt#3\fifo}
%a blank line
\fifo\def\⟨name1⟩{...}
     \def\⟨name2⟩{...}
     ...
     \def\⟨namen⟩{...}
\ofif\relax\relax
\bye
```

Note the difference.

6.3 blue.tex contains a macro `\contentsdatabase` that writes the list of names of a lit, pic, or address database to the file contentslit, contentspic, or contentsaddress.
Let us assume that we wish the list of names in a toks variable `\namelst`. To facilitate processing of `\namelst` `\lst` has been inserted before each name in the list.

```
\newtoks\namelst
\def\lst#1#2{\namelst\ea{\the\namelst\lst#1}}
\input lit.dat
%and to verify the contents
\baselineskip12pt plus2pt minus1pt
\lineskip1pt plus 1pt minus1pt
\def\lst#1{{\tt\string#1}\endgraf}
\the\namelst
```

6.4 The number 1. Once the reference has been found, it will be used, and the name will be overwritten by its sequence number.

7.1 As usual it all depends on your application. End of answer. But—there is always a but—the complaint I heard most about *The TEXbook* was that the information is spread all over, and that it is hard to find what you are looking for. Therefore I consider a few page numbers per entry beneficial. (Let us forget about the intrinsic complexity of the subject, certainly at the time.) BLUe's format supports scrutinizing parts of an index, because it is so easy to generate an index per chapter on the fly. It is hardly not more difficult than generating a table of contents. An index per chapter can be scrutinized more easily, and redundancies removed. That the index provides a mechanism to link things over chapters is a good thing, however. Don't misunderstand me. But don't overuse it, IMHO, with all respect. Remember DeVinne's adage 'The last thing to learn is simplicity.'

7.2 The naive approach is to enclose that part by braces and precede it by a circumflex. However, that goes wrong because a title is stored and reused in many places. So copy the words and mark them as a silent IR.

7.3 Add an extra level of braces.

8.1 The regular headlines and footlines are as above with a few elements altered. However, there is a complication in the sense that the headline is suppressed for the first page of the chapter. This is handled in `\postchapterhead`. Consult the formatbase, especially the report format.

8.2 Similar to the running head of a book the headline—well, the header part—can be put in a headline with contextual information dynamically coupled. See the formatbase, especially the transparencies format.

8.3 Provide

```
\footline{\hbox to\pagewd{\ifodd\pageno\null\hfill\fi\number\pageno\hfil}}
```

8.4 Let `\makesignature` equal to `\relax`.

8.5 The token variables `\⟨tag⟩name`: `\abstractname`, `\keywordsname`, `\contentsname`.

8.6 The control sequence to obtain Roman numerals is `\romannumeral`. In plain TEX Roman numerals are obtained automatically for page numbers when `\folio` is used instead of `\pageno` provided the number is negative. From *The TEXbook* 252, 362 the following.

```
\def\folio{\ifnum\pageno<0 \romannumeral-\pageno
           \else\number\pageno \fi}
```

See also page 40 and 41 of *The TEXbook*, especially when uppercase roman numerals are needed. In blue.tex I introduced the token variable `\pagenumber{\folio}`.

8.7 This is a very common TEXing paradigm. The idea is to redefine the responsible control sequence in such a way that on first invocation the only action is to define itself by the regular replacement text. For example in the report format, in order to suppress the headline on the first page of each chapter, I defined `\headline` in `\postchapter` as follows.

```
\postchapterhead={\vskip3pc\headno=0 \exno=0 \exano=0
   \headline={\global\headline={\hbox to\pagewd
    {\sevenrm\the\chaptername\hfill\it\the\title}}}}
```

8.8 The explanation is in the definition of `\notlastscript` and `\lastscript`, which read as follows.

```
\def\notlastscript{\let\stop\relax}
\def\lastscript{\let\stop\end}
```

B: Formats

Next to the regular format BLUe's format contains the report and transparencies formats. The letter and concert format from the TEXbook have been reviewed and the goodies in there have been borrowed. The letter format makes use of the address.dat database. Treated are among other things how to use the formats, how to proof separate chapters of a report, how to create and update a set of transparencies, how to format letters, make address labels, and how to extend the databases by a format, tool, or address of your own. A table of main tags and a list of the Frequently Used Tags, for the letter, report and transparencies format, have been included.

Before delving into the details of the database approach, I'm happy to let you know that you can do without it, can fall back on the traditonal method of file inclusion, if for some reason or other you don't feel comfortable with databases. For example, if you like to make use of the transparencies macros from BLUe's format, copy the relevant part from fmt.dat into, say, trspar.tex. Then proceed as usual, for example as follows.

```
\input blue.tex
\input trspar.tex
⟨restofbluescript⟩
```

On the other hand those who like the database approach can peruse 'BLUe's Format databases.'

Use of formats from fmt.dat database

To use a format just start a script with[85]

```
\input blue.tex \⟨formatname⟩
```

The markup tags are upwards compatible, i.e., the tags from the regular blue.tex are still there, but extended with extra tags. The coding of the tags from the default format have been adapted to the look-and-feel of \⟨*formatname*⟩, the design if I may say so.

Adding a format To add a format to the BLUe's format system the following has to be done.

- separate the storage-allocating control sequences from the format, i.e., treat `\newtoks` and the like separately
- add the allocation control sequences to blue.tex
- add the format with an 'opening' and 'closing' to fmt.dat
- add a load macro to blue.tex.

In actual fact this comes down to the following. Add to fmt.dat

```
\tool\⟨yourformat⟩fmt%Opening
⟨themacros⟩
\endinput             %Closing
```

[85] Inclusion of files etc. is handled behind the scenes. Default the format is suited for an article.

and include in the kernel blue.tex the load macro and the storage allocations

```
%load macro
\def\⟨yourformat⟩{\let\⟨yourformat⟩fmt=x
\loadformat
\let\⟨yourformat⟩fmt=\undefined
}
%storage allocations
\new⟨type⟩\⟨name⟩
%⟨type⟩ is  one of box count dimen fam insert language skip
%                   muskip read toks write
%et cetera
```

Note the difference between `\`⟨*yourformat*⟩ and `\`⟨*yourformat*⟩`fmt`. Take care that `\`⟨*yourformat*⟩`fmt` remains undefined afterwards. Now and then I safeguarded formats from double loading.

There are a few restrictions on the macros, as a consequence of the skipping mechanism used, via TEX's argument processing.

- no outer defs, i.e., `\newtoks` and the like should be declared in regular blue.tex
- no `\par`, use `\endgraf`, or `\csname par\endcsname`
- no blank lines, insert %

Remarks. The restriction with respect to your outer defs can be circumvented by

- define all macros non-outer, or
- use `\csname newtoks\endcsname\`⟨*tag*⟩, and the like, or
- extend `\unouterdefs` by your outer defs, i.e., include `\newtoks` and similars. When these are part of the format or tool, multiple use reallocates memory, and because TEX lacks a garbage collector this will cost extra memory locations.

Check for data integrity After a format has been added one can check whether the file can be scanned by TEX via making a table of contents as follows.

```
\input blue.tex
\contentstoolsorfmt{fmt}
\bye
```

This script produces no pages of output (that is OK). The file contentsfmt will be created though.

Verbatim listing of a format This problem is induced by my multi-file approach. Of course one can extract the format by an editor et cetera as alternative.

Example *(Verbatim listing of letter format)*

```
\input blue.tex \let\letterfmt=x
\long\def\tool#1{\ifx#1\undefined\bgroup\unouterdefs\ea\gobbletool
   \else\ea\toolverbatim\fi}
%\def\toolverbatim{\thisverbatim{\baselineskip12pt plus2pt minus1pt
%   \lineskip1pt plus1pt minus1pt}\beginverbatim}
\def\cgl{\endverbatim\endinput}
\input fmt.dat \bye
```

Conventions for entries fmt.dat come down to

- update the contents at the beginning of the file fmt.dat
- supply `\message{ ---`⟨...⟩ `format,` ⟨*date*⟩`,` ⟨*owner*⟩`--- }` in the beginning
- take care of visual layout for print out
- provide additional comments which contain markup for line numbering
- add history of changes
- add a table of contents with line numbers.

The layout is aimed at printing in two columns and starts in a new column. The selective line numbering is controlled by the following control sequences

`\numvrb`, to start line numbering

`\vrblin` = ⟨*number*⟩, to adjust line counter, and

`\nonum` to switch off line numbering.

Do remember that the escape char is the semi-colon.

Example *(Template for an entry in fmt.dat)*

```
%end%%%%%⟨lastlinepreviousformat⟩%;newcol
%
%
%begin%%%%%%%%%%%%%%%%%%%%⟨....⟩%;numvrb
\tool\⟨...⟩fmt
\message{ ---⟨...⟩ format, ⟨date⟩,
                         ⟨owner⟩--- }
%⟨authorinformation⟩
%⟨theformatproper⟩
\endinput                            %;nonum
%Contents
%⟨item⟩...................⟨pagenumber(s)⟩
%etc
%History of changes
%⟨date⟩ ⟨change⟩
%etc
%end%%%%%%%%%%%%%%%%%%%%%%⟨....⟩%!cgl;newcol
```

Concert format

Please open *The TEXbook* on page 411. What about this concert announcement? I never received such a leaflet when attending a concert. The ones I got are prettier, so there is no use in copying this template.[86] This is more or less in agreement with what Knuth states

> 'Notice that the `\composition` and `\movements` and `\soloists` macros do not include any special provision for vertical spacing; the user is supposed to insert `\smallskip`, `\medskip`, and `\bigskip` as desired. This was done deliberately, because different concert programs seem to demand different spacing; no automatic scheme actually works very well in practice, since musical literature is so varied.'

What can we learn from the format, especially the macros? Usually a lot. Here we go:

- The simple size-switching macros are nice, intelligible, and compact, because we don't deal with math in this context. The tags are just `\smalltype`, `\medtype`, and `\bigtype`. A pedagogical stepping stone for size-switching macros in general.
- The `\tsaologo` is nice too. I have used it for delineating parts in 'Math into BLUes.' My variant is the macro `\partlogo` with the text as argument. The latter is part of blue.tex.
- The `\soloists` macro deserves more general use. It shows how to use `\halign` to *center a block of text*. Moreover, it can be recast into a two-part macro. Note that `\centerline` has been used, but the `\vbox` could have been replaced by math display markup with the automatic insertion of `\displayskip` before and aft. And why not get rid of the `\cr`? Natural input? A matter of taste? I agree partly. The advantage is the extra option possibility after the opening keyword. In this case we can parameterize over the font to be used.

[86] I have included the macros in the format `\concert` nevertheless, for your convenience.

Centering a block The coding of `\soloists` can be combined with the coding of `\begindisplay`, *The TEXbook* 421, to yield a two-part macro for centering a block, with the flexibility to specify the font after `\begincenterblock`. Moreover, I incorporated the natural markup idea. No `\cr` has to be supplied by the user, TEX will insert those. The macro is part of blue.tex.

```
\def\begincenterblock{\bgroup
 \let\par\cr\obeylines\startcenterblock}
{\obeylines\gdef\startcenterblock#1
  {$$#1\vbox\bgroup\halign\bgroup&##\hfil\cr}}
\def\endcenterblock{\egroup\egroup$$\egroup}
```

Example *(The soloists macro; block centering)*

```
\begincenterblock\bf
Ben Lee User, horn
Peter Schickele, hardart
\endcenterblock
```

Ben Lee User, horn
Peter Schickele, hardart

Remark. The disadvantage of this macro is that no page break will occur within the vbox.

Letter format

Please open *The TEXbook* on page 403, and you will see a marvel of natural markup. Only 5 tags are needed to mark up a letter.

`\letterhead \address \body \closing \endletter`

If this is such a great approach, how come that it has not been taken over? IMHO, with all respect, it was too difficult to customize. At the time TEXies were busy porting the system to the various platforms. Moreover, nowadays I would like to

- store addressees in a database, and use them from there
- have a simpler letterhead
- let `\headline` take care of the headers
- use fewer fonts, prevent suffering from fontitus
- use the known sender—remember BLUe's format is yours, it knows about your context
- use window envelopes
- introduce token variables for `\subject`, `\yourreference`, and `\ourreference`
- make positioning of the address flexible in such a way that it can move to adjust to your window envelope
- adapt `\makelabel`[87] to work independenly with the address.dat database (Handy for address labels for carbon copies among other things)
- substitute `\beginscript` for `\beginletter` and ditto for the `\endscript`.[88]

Because of the above, it is of no use to include the letter format literally from *The TEXbook*. Furthermore, I simplified the approach a little, :-))), to facilitate customization. The variant has been incorporated as `\letter` format in the fmt.dat database.

Note. The `\footline` on the first page is free to contain business information like giro, Chamber of Commerce number and the like. BLUe's format contains a toks variable `\businessaccount` for that purpose. See NTG's letter format.

[87] I named it `\makelabels`.
[88] Well, . . . I allowed `\beginletter` as alias, ditto for `\endletter`.

Use of the letter format In order to use this format the user has to customize his letter format and make sure address.dat is available.[89]

```
\input blue.tex \letter %Knows about logo, sender
\subject{...}
\yourreference{...}     \ourreference{...}%just numbers
\addresses{\⟨name⟩}     %load list of address{es}
\addressee{\⟨name⟩}
\beginscript
\dear
....
\sincerely              %knows about you
\cc ...                 %Optional set cc. ....
\ps ...                 %Optional set P.S. ....
\appendix{⟨appendixtitle⟩}
⟨appendixmaterial⟩
\endscript
```

The `\appendix` control sequence takes one argument. It starts a new page but continues numbering.

If the letter proper is stored in letter.tex, then the use looks even simpler.

```
\input blue.tex \letter          %Knows about logo, sender
\subject{...}
\yourreference{...}
\ourreference{...}               %just numbers
\letterto{\⟨name⟩...}
```

Example *(A letter script; a letter to Don Knuth)*

```
\input blue.tex \letter          %Preliminary initializations
\subject{\TeX{} for BLU}
\ourreference{22 1 95}
\yourreference{\TB}
\addreesses{\knuthde}
\addressee{\knuthde}             %Initializes \addresseename \affiliationbox
                                 %\thisscript{\notlastscript}
\beginscript                     %\beginletter is an alias
\dear
First of all I'm  much obliged to your works especially \TB.
As you know roughly 10k organized users profit from  your system
in getting their work out. It is estimated that the number of
{\sl unorganized\/} users is a tenfold.
I like your Ben Lee User character very much,
and hope he will benefit from my work towards understanding
and using \TeX{} with confidence.
\bigskip
What puzzles me is that manmac has been neglected so much.
Is it because a user's guide was lacking?
Was \TeX{} too far ahead of its time?
Too unusual?
\bigskip
I'm also puzzled about the neglectance of the letter format.
Was it too difficult for \TeX ies to customize at the time?
I mean, it is confusing to mix the line mode and the par mode.
As you can see from the \cs{letter} format I included, as part of
BLUe's format, I refrained from those and concentrated on
customization.
```

[89] Well, the latter can be circumvented if `\loadreference` is replaced by a def with the name of the addressee, followed by the affiliation of which each line is preceded by `\\`, as replacement text.

```
I also made use of a database of addresses, but that is
a minor \TeX nical point, because the markup abstracts from
that detail as it should. Just \adressee\⟨name⟩.
\bigskip
Whatever the reasons, I hope that my work will ease the use of manmac,
or even better BLUe's format.
The latter has manmac assimilated, and taken notice of
experience gained since the release of manmac.
\bigskip
After this work on using \TeX{} I intend to write
another booklet on \TeX{} macro writing, to emphasize two-part macros,
because two-part macros are in the dark in the \TeX book.
They are lacking in the chapter on macro writing, but
have been used in manmac, and therefore are dealt with in Appendix~E.
IHMO, with all respect, they deserve more attention.
By that time, I hope to be familiar with `\TeX{} The Program' too.
\bigskip
In near future I'll direct myself  to \MF, well META{\smc post},
and pay attention to 2.5-D applications.
I have in mind to emulate the works of Naum Gabo, a sculptor,
whose objects I like very much. The exciting thing in there is
that I can emulate the work in 2.5-D with only 1-D information.
The latter is quite obvious because he used regular surfaces.
However, it is apparently me who arrived at the crossroads.
\bigskip
I would appreciate it very much if you would let me know
that you have received this work, and of course,
I welcome suggestions.
\sincerely
\ps When I entered `Het Nederlands Rekenmachine Genootschap,'
    in {\oldstyle1967}, the first volume of the Art of Computer
    Programming was given to me, because of my lucky membership
    number, {\oldstyle1024}.
    I really could not foresee at the time that my interest in your work
    would take this turn
\cc NTG
\endscript
```

Nederlandstalige
TEX
Gebruikersgroep

Hunzeweg 57
9893 PB Garnwerd
The Netherlands
cgl@rc.service.rug.nl
31 5941 1525

D.E Knuth
Department of The Art of Computer Programming
Stanford University
Stanford CA 94305
USA

Subject : TEX for BLU
Our Ref : cgl 22195
Your Ref : *The TEXbook*

March 9, 1995

Dear Prof. Knuth,

First of all I'm much obliged to your works especially *The TEXbook*. As you know roughly 10k organized users profit from your system in getting their work out. It is estimated that the number of *unorganized* users is a tenfold. I like your Ben Lee User character very much, and hope he will benefit from my work towards understanding and using TEX with confidence.

What puzzles me is that manmac has been neglected so much. Is it because a user's guide was lacking? Was TEX too far ahead of its time? Too unusual?

I'm also puzzled about the neglectance of the letter format. Was it too difficult for TEXies to customize at the time? I mean, it is confusing to mix the line mode and the par mode. As you can see from the `\letter` format I included, as part of BLUe's format, I refrained from those and concentrated on customization. I also made use of a database of addresses, but that is a minor TEXnical point, because the markup abstracts from that detail as it should. Just `\adressee\`$\langle name\rangle$.

Whatever the reasons, I hope that my work will ease the use of manmac, or even better BLUe's format. The latter has manmac assimilated, and taken notice of experience gained since the release of manmac.

After this work on using TEX I intend to write another booklet on TEX macro writing, to emphasize two-part macros, because two-part macros are in the dark in the TEXbook. They are lacking in the chapter on macro writing, but have been used in manmac, and therefore are dealt with in Appendix E. IHMO, with all respect, they deserve more attention. By that time, I hope to be familiar with 'TEX The Program' too.

In near future I'll direct myself to METAFONT, well METAPOST, and pay attention to 2.5-D applications. I have in mind to emulate the works of Naum Gabo, a sculptor, whose objects I like very much. The exciting thing in there is that I can emulate the work in 2.5-D with only 1-D information. The latter is quite obvious because he used regular surfaces. However, it is apparently me who arrived at the crossroads.

I would appreciate it very much if you would let me know that you have received this work, and of course, I welcome suggestions.

Sincerely,

Kees van der Laan

P.S. When I entered 'Het Nederlands Rekenmachine Genootschap,' in 1967, the first volume of the Art of Computer Programming was given to me, because of my lucky membership number, 1024. I really could not foresee at the time that my interest in your work would take this turn

cc. NTG

Syntax entries address database The address.dat database contains addresses sorted on name. Each entry obeys the syntax, as shown by the following example.

```
\lst\⟨name⟩{⟨salutationname⟩
  %Each line preceded by \\
  \\⟨fullname⟩
  \\⟨affiliation⟩
  \email{...}%optional
  \phone{...}%Optional
}
```

Example *(Entry address.dat database)*

```
\lst\ntg{NTG
\\Nederlandstalige \TeX{}
          Gebruikersgroep
\\Postbus 394
\\1740 AJ Schagen
\\The Netherlands
\email{ntg@nic.surfnet.nl}
}
```

To add an address just adhere to the syntax, and include the new address in address.dat, in the right order.

Check for data integrity After an address has been added one can check whether the file can be scanned by TEX via making a table of contents as follows.

```
\input blue.tex
\contentsdatabase{address}
\bye
```

This script produces no pages of output (that is OK). The file contentsaddress will be created though. If you have a file with addresses of your own, say abc.dat, provide also `\newwrite \tocabc`.

Multiple letters Erik Frambach asked me how to format multiple letters, with each letter personalized by incorporating the address. If we are talking of a few letters there is no problem at all. Store the letter in a file and repeat the formatting for each letter.

Let us assume that we wish to address a list of addressees, and that these addressees are already in our database address.dat.

The approach to solve this is to go through the list of adressees, via FIFO, and 'process' each with the letter.

```
\input blue.tex \letter
\subject{\TeX{} for BLU}
\yourreference{\TB}
\ourreference{1$\,$2$\,$95}
\letterto{\knuthde\ntg}
\bye
```

This scheme also works for one letter. Note that after Dear the salutation name is used, which gives a personal flavour.

And what about a letter to the complete address database? Logically, this is simple. Make a list of all the names of the database elements and supply this list to `\letterto`. We can let TEX do all the work, by using `\lettertoall` as follows.

```
\input blue.tex \letter
\subject{\TeX{} for BLU}
\yourreference{\TB}
\ourreference{1 2 95}
\lettertoall
\bye
```

An interesting application is that for example NTG can send a—more or less personalized—letter to all its members.

Address labels Especially for the case of carbon copies separate address labels are wanted. For that purpose I modified Knuth's `\makelabel` into `\makelabels`.

```
\input blue.tex \letter
\makelabels{\knuthde\ntg}
%and all address labels
%can be obtained via
\makealllabels
\bye
```

If you prefer smaller addrees labels provide `\twocol`.

Searching the address.dat database by pattern BLUe's format provides a `\search` macro with a search pattern as argument. The purpose is to search the database file provided in the toks variable `\searchfile`, for this search pattern. The result of the search is a list of names, of which the replacement text contains the search pattern. This list of names is delivered in the toks var `\namelist`, and for the time being they are also written to the log file.

Example *(Search for addresses from RUSSIA in the database address.dat)*

In order to justify my use of the word database in relation to TEX, I provided the search macro to browse the databases lit.dat and address.dat. Below the input and result have been given of the search of address.dat for the pattern RUSSIA. The result is contained in the toks variable `\namelst`.

```
\input blue.tex
\searchfile{address}
\search{RUSSIA}
\bye
```

```
\lst\gorbunovai\lst\grinevaoa\lst\makhovayai
```

This is handy for making address labels grouped per country.

```
\input blue.tex \letter
\search{RUSSIA}
\makesearchlabels
\bye
```

Do you see the extension for sending letters to all those people?

Remark. A search pattern should not contain a period.

Customization If there is one format which is candidate for customization this is the one. However, much information is already known by the system, like `\author`, `\address`, `\logo`, et cetera. Also addressees are known because of the address database. What is left is to position the address in the address window. BLUe's format provides the skip variables `\haoffset` and `\vaoffset`. These determine the shifts of the address within the window, horizontally or vertically, respectively.

When you are not satisfied with the basic layout then the `\headline` and `\footline` have to be rewritten, because how the information is pasted up will be taken form these token variables. How to design your own letter format is beyond the scope of this user's guide.

Your summary of main markup tags for letters The Frequently Used Tags, FUTs for short—indeed a handful, as already indicated in the template in the 'Introduction'—for personalizing. Use it as crib.

Letter proper

```
\addresses{...}
\addressee{...}
\subject{...}
```

```
\yourreference{...}
\ourreference{...}%only digits
\letterto{...}
\lettertoall
\vaoffset
\haoffset
```

Copy Proper

```
\beginscript
\endscript
\dear
\sincerely
\cc
\ps
\appendix{...}
```

Address labels

```
\makelabels{...}
\makealllabels
\makesearchlabels
```

Report format

A report is set in one-column, with per page a header and footer. The header is suppressed for the preliminary pages and for every first page of a chapter. The latter starts with a title and summary followed by the copy proper. The footer is suppressed for the preliminary pages, i.e., these pages lack numbering, among other things.

Some new control sequences are

`\preface`	for a preface
`\chapterhead`	for chapter title
`\setupappendices`	for letters as numbers
`\pasteup...`	to paste up elements

Next to the coding of the new control sequences several existing control sequences had to be recoded to comply with `\report`. The reimplementation of `\beginscript` was tedious work, because it takes care of the preliminary pages: the cover page, the inside cover page, the title page, the preface page, and the table of contents page. `\endscript` takes care of the back cover pages.

The implementation of `\(chapter)head` was tough work because of the wish to process on the fly and to store the title. On second thoughts I realized the more modest wish to allow for math in the titles, especially to handle the circumflex appropriately.

Next to the a priori and physical table of contents in the front matter of the report, a table of contents with page numbers attached dynamically, is generated on the fly and appended at the end of the typeset report.

Template The file for this report has all chapters included. However, while writing and proofing all chapters were kept in separate files. The latter structure, stripped from redundancies, has been included below as a template.

If the preliminary matter is a little overwhelming, skip this section, and continue with ‘Proofing a single chapter.’

Example *(A report script; the outer markup for this report)*

The template reads as follows.

```
\input blue.tex %\loadindexmacros  %When creating an index too
\report
                                   %Cover data
\bluetitle Publishing with \TeX

\bluesubtitle BLUe’s Selection

\bluepictures \bluemodelpic

                                   %Inside cover data
\bluekeywords Bibliography, \bluetex, ...

\bibliographydata{ISBN: \dots\par ...}
\beginacknowledgements
First of all
\displaycenterline{Don Knuth, Thank You!}
...
\rightline{I’m grateful to all of you.}
\endacknowledgements
                                   %Title page data (next to title&author)
\beginabstract
The use of \bluetex{} has been documented in a user’s guide,
...
\endabstract
                                   %Preface page data
\beginpreface                      \vskip2\bigskipamount
```

```
Dear User,                         \bigskip
It occurred to me that my detailed studies
...
will be beneficial for new-and-old \AllTeX ies alike.
                                   \bigskip
\rightline{\the\author}
\endpreface
                                    %Contents Page data
                                    %Because of \valign use, \contents{...}
                                    %is not possible.
\begincontents                      %\se is subentry descriptive tag
Preface
Table of contents
Introduction
\se BLUe's format
...
\se J: Table of Contents
\endcontents
\let\coverpic=\bluemodelpic         %BLUe script proper
\beginscript
\input pwt.int                      %Introduction
\input pwt.txt                      %Text
%\endscript %For testing, to make script smaller
\input pwt.mat                      %Mathematics
\input pwt.tab                      %Tabular material
\input pwt.grf                      %Pictures
\input pwt.ref                      %References
\input pwt.ind                      %Index creation
\input pwt.cus                      %Customization
\input pwt.aft                      %Afterthoughts
%
\setupappendices                    %From now on A, B, C,...
\pasteupanswers                     %A: Answer to Exercises
\input pwt.fmt                      %B: Formats
\input pwt.tls                      %C: Tools
\input pwt.sur                      %D: Backgrounds
\pasteuptoe                         %E: Table of Examples
\input pwt.tot                      %F: Table of tags
\input pwt.his                      %G: History of changes
\input pwt.obs                      %H: Obselete macros
%\sortindex\pasteupindex            %I: Index
\pasteuptoc                         %J: Table of Contents
\endscript
```

Proofing a single chapter Chapters can be proofed separately as follows.

```
\input blue.tex                     %\loadindexmacros
\report                             %After \loadindexmacros
\input ⟨filewithchapter⟩            %\sortindex\pasteupindex
\bye
```

Perhaps the easiest way to get results is to forget about the preliminary matter to start with. The above script formats your chapter and why not begin with `\chapterhead{...}` right away, instead of `\input` ⟨*filewithchapter*⟩?

We can easily test a few chapters in a row as follows. Omit `\bye` and when TEX prompts for an `\end` insert 'i' for insert and after the insert prompt supply `\input` ⟨*nextchapter*⟩.

Main tags for report format A table of main tags for `\report` has been included to provide a survey of what outer tags—and to what extent—they have been implemented for the `\report` format.[90] The table also illustrates the naming convention of a root name with affixes.

The header fields indicate, the prefixes pre-, post-, this-, every-, pasteup-, and the suffixes -font, -name and -box. The first column contains the various root names, with a marker to denote to what extent the minimal one-part macro has been implemented.

Main tags for report scripts

Root \ Affix		pre-	post-	this-	every-	pasteup-	-font	-name	-box
abstract	+	¬	¬	¬	¬	+	¬	+	+
acknowledgements	+	+	+	+	¬	+	¬	+	+
btable	▯	+	+	+	+	¬	¬	¬	+
center	+	¬	¬	¬	¬	¬	¬	¬	¬
chapterhead	+	+	+	+	+	¬	+	¬	¬
contents	+	¬	¬	¬	¬	+	¬	+	+
(sub(sub))head	+	+	+	¬	¬	¬	+	¬	¬
keywords	+	¬	¬	¬	¬	+	¬	+	+
picture	¬	¬	¬	+	+	¬	¬	¬	¬
pictures	▯	¬	¬	¬	¬	¬	¬	¬	¬
preface	+	+	+	¬	¬	¬	+	¬	¬
quote	+	¬	¬	¬	¬	¬	¬	¬	¬
references	▯	+	+	+	+	+	¬	+	+
script	+	+	+	¬	¬	¬	+	¬	¬
summary	+	¬	¬	+	+	¬	¬	¬	¬
verbatim	¬	+	+	+	+	¬	¬	¬	¬

¬ not implemented
\+ implemented
▯ implemented without processing on the fly

Remarks

- `\title`, `\subtitle`, `\author` and the like, are token variables and associated with the script
- `\pictures` and `\references` have been implemented in a straightforward way, because their use doesn't need processing on the fly.

Your summary of main markup tags for reports The Frequently Used Tags, FUTs—indeed a handful, as already indicated in the template in the 'Introduction'—for personalizing.

Preliminary matter
```
\title{...}
\subtitle{...}
\keywords{...}
\abstract{...}
\contents{...}
```

[90] More control sequences and symbols are available, especially for the lower level markup, if not for the undocumented features of manmac, and the wealth of plain TEX.

```
\issue{...}
\acknowledgements{...}
\references{...}
\pictures{...}
```

Copy Proper

```
\beginscript
\endscript
\chapterhead{...}
\summary{...}
\head{ ...}
\thissubhead{...}
\subhead{...}
\thissubsubhead{...}
(the \this⟨headkind⟩{...}  especially together with \runintrue)
\subsubhead{...}
```

Inner level

```
\begindemo...⟨escapechar⟩yields...\enddemo
\thisverbatim{...}
\beginverbatim
⟨escapechar⟩endverbatim
(and in-line | ... |)
\beginquote
\endquote
\begindisplay
\enddisplay
(with & and \cr)
\displaycenterline{...}
\begincenterdisplay
\endcenterdisplay
(with & and \cr)
\btable
(with \data{...\cs...\rs...\cs...\rs...}
      \header{...}
      \footer{...}
      \rowstblst{...}
      \vruled{...} )
\ref\⟨name⟩
\crsref\⟨name⟩
\ftn{...}
\item{...}
\itemitem{...}
\aitem
\Aitem
\bitem
\nitem
\example{...}
\thispicture{...}
(with \unitlength=...)
\cs{⟨tagname⟩}
```

Back matter

```
\setupappendices
\pasteupanswers
\pasteuptoe
\pasteupreferences
\sortindex\pasteupindex
\pasteuptoc
```

Addition of a format to BLUe's format system

Example *(Addition of report format to the fmt.dat database)*
The following has been added to fmt.dat.

```
\tool\reportfmt
⟨themacros⟩
\endinput
```

blue.tex has been extended with

```
\def\report{\ifx\undefined\reportfmt\let\reportfmt=x\fi
   \ifnum\reportloadcnt=0 \ea\loadformat\fi
   \advance\reportloadcnt1 \let\reportfmt=\undefined}
%Storage allocations for \report come next
\newtoks\prefacename
%et cetera
```

Transparencies format

BLUe's format transparencies have a centered flavour, with a header part which reflects the series, and the context. There is also room for a user logo to be printed in the header.

Template I like to start from an example, a template, which shows what has to be provided.

Example *(Transparencies script)*

```
\input blue.tex \transparencies \vfuzz=15pt \everyverbatim{\emc}

\title{BLUe's Format}\subtitle{the best of both worlds}

\contents{WAF?
 Outer vs Inner markup
 Template
 Customization
 Coding
 Transparencies
 Tags survey
 Conclusions
}% end contents

\pictures{\bluemodelpic}

\beginscript
\head{WAF? }                        %Starts a new transparency, implicitly
$$\thispicture{\xoffset{-1}}\bluemodelpic$$

\head{Markup language}              %Starts a new transparency, implicitly
\bitem \begin⟨tag⟩  \end⟨tag⟩  \⟨tag⟩   \hfill defs
\bitem \this⟨tag⟩ \every⟨tag⟩\hfill token vars
\bitem \cs{title} ...              \hfill token vars
\bitem \pre⟨tag⟩    \post⟨tag⟩   \hfill token vars
\bitem \pasteup⟨tag⟩                \hfill defs
\smallskip
\bigskip
\beginverbatim
\⟨tag⟩      or      \begin⟨tag⟩
                      ...

                     \end⟨tag⟩

%and

\every⟨tag⟩{...}     \this⟨tag⟩{...}

\pre⟨tag⟩{...}       \post⟨tag⟩{...}
!endverbatim
\loadxyz
$$\xyz{Alias tags}{Options}{Place within context}$$
\endscript
```

The following pages contain the results. `\transparencies` has been used locally, and because of the nested use I inserted `\thisscript{\notlastscript}`.

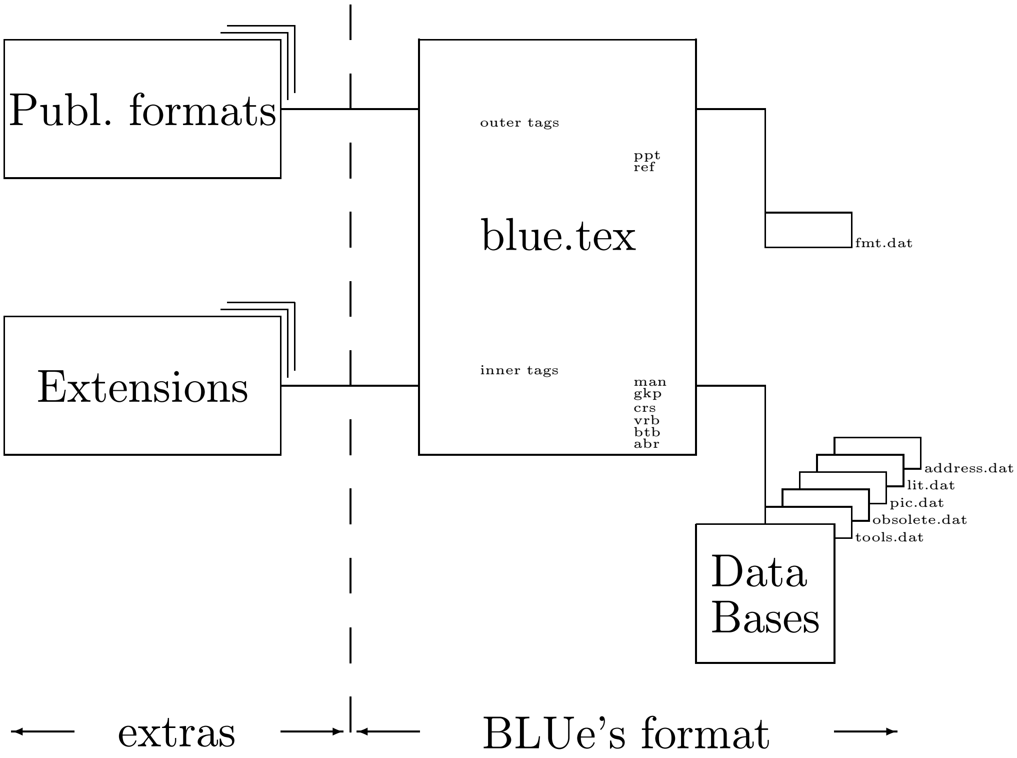
BLUe's Format

—the best of both worlds—

Kees van der Laan
Hunzeweg 57
9893 PB Garnwerd
The Netherlands

- WAF?
- Outer vs Inner markup
- Template
- Customization
- Coding
- Transparencies
- Tags survey
- Conclusions

WAF?


Publ. formats
Extensions
outer tags
ppt
ref
blue.tex
inner tags
man
gkp
crs
vrb
btb
abr
fmt.dat
address.dat
lit.dat
pic.dat
obsolete.dat
tools.dat
Data
Bases
extras
BLUe's format

Markup language

- `\begin`$\langle tag\rangle$ `\end`$\langle tag\rangle$ `\`$\langle tag\rangle$ — defs
- `\this`$\langle tag\rangle$ `\every`$\langle tag\rangle$ — token vars
- `\title` … — token vars
- `\pre`$\langle tag\rangle$ `\post`$\langle tag\rangle$ — token vars
- `\pasteup`$\langle tag\rangle$ — defs

`\`$\langle tag\rangle$ `or` `\begin`$\langle tag\rangle$

. . .

`\end`$\langle tag\rangle$

`%and`

`\every`$\langle tag\rangle$`{...}` `\this`$\langle tag\rangle$`{...}`

`\pre`$\langle tag\rangle$`{...}` `\post`$\langle tag\rangle$`{...}`

Place within context

Options

Alias tags

Use your logo `\logo` is a def, and invoked within `\line`. In order to have your logo, supply

```
\def\logo{\vbox to0pt{\vss
...              %Your logo
}}
```

Maintenance This section is in a certain sense artificial, introduced by the automatisms which come with my choices for the 'header' and 'footer,' especially the context-dependent information in there. The following might be helpful for modifying or extending single transparencies. The idea is to follow the script setup and handle the `\header`, `\pagenumber` and `\identification` for the particular transparency. In the example below the task was to add a transparency in between.[91]

```
\title{Literate Programming}
\beginscript
\head{Tools}
\nxttrs
\identification{feb94}
\pagenumber{13a}
\subhead{FWEB}
LP tool by John Krommes
\endscript
```

Main tags for transparencies A table of main tags has been included to provide a survey of what outer tags—and to what extent—they have been implemented for the `\transparencies` format. The table also illustrates the naming convention adopted via a root name with prefixes and suffixes.

The header fields indicate, the prefixes pre-, post-, this-, every-, pasteup-, and the suffixes -font, -name and -box. The first column contains the various root names, with a marker to denote to what extent the minimal one-part macro has been implemented.

Main tags for transparencies scripts

Root \ Affix		pre-	post-	this-	every-	pasteup-	-font	-name	-box
btable	▯	+	+	+	+	¬	¬	¬	+
center	+	¬	¬	¬	¬	¬	¬	¬	¬
contents	+	¬	¬	¬	¬	+	¬	+	+
(sub(sub))head	+	+	+	¬	¬	¬	+	¬	¬
picture	¬	¬	¬	+	+	¬	¬	¬	¬
pictures	▯	¬	¬	¬	¬	¬	¬	¬	¬
quote	+	¬	¬	¬	¬	¬	¬	¬	¬
script	+	¬	¬	+	+	¬	¬	¬	¬
verbatim	¬	+	+	+	+	¬	¬	¬	¬

¬ not implemented
\+ implemented
▯ implemented without processing on the fly

Remarks

- `\title`, `\subtitle`, `\author` and the like, are token variables and associated with the script
- `\pictures` has been implemented in a straightforward way, because it doesn't need processing on the fly.

[91] Without automatically gathering of context information the modifications are a trifle.

Your summary of markup tags for transparencies

Token vars

```
\address\author\crowner%Default, already personalized
\every...              %See default blue.tex
\footline
\headline
\headtitle
\identification
\pagenumber
\posthead
\postsubhead
\postsubsubhead
\prehead
\presubhead
\presubsubhead
\subheadtitle
\subsubheadtitle
\subtitle
\this...%See blue.tex
\title
```

Control sequences with root name

```
center %default
contents
head
pictures
script
```

Auxiliaries

```
\continue
\nxttrs
```

Lower level

```
\bitem
\beginverbatim
```
⟨*escape char*⟩`endverbatim`

For inner markup the macros from plain and those from blue.tex can be used.

Do remember the line-by-line character of the script

C: Tools

BLUe's format comprises tools.dat, a database of tools, available at your fingertips, which doesn't hinder when not needed.

At the moment tools.dat, the database of tools, contains macros for typesetting bridge, crosswords, the tower of hanoi game, creation of an index (and in there sorting macros), ntglogo, and Pascal. Most likely this database will grow.

Use

Nobody would be surprised to hear that the ease of use of a tool has all to do with the tool's jargon as implemented by its author. However, how to access the tool is also important, and we would love a uniform approach to that, wouldn't we? Alas, consistency at the outer level—how the tools can be accessed from the database tools.dat—is too heavy a demand, as can be seen from the index tool in relation with the other tools. The tools are just too varied.

Adding a tool First we have to decide what it is that we like to add. A format? A picture? Or just a tool? I had to add the `\xyz` picture to tools.dat, because it is not a format and it does not obey the syntax of pic.dat entries, although it looks like a picture in print to me.

Remember that the entries in the lit.dat, pic.dat, and address.dat database obey the syntax of a def, *without parameters*.

`\def\`⟨*name*⟩`{`⟨*replacement text*⟩`}`.

The entries of fmt.dat and tools.dat obey the syntax

`\tool\`⟨*name*⟩`fmt`		`\tool\`⟨*name*⟩`tool`
`...`	or	`...`
`\endinput`		`\endinput`

To add a tool the following has to be done.

- separate the storage-allocating control sequences from the tool, i.e., treat `\newtoks` and the like separately
- add the storage-allocating control sequences to blue.tex
- add the tool as a definition to tools.dat
- add a load macro to blue.tex.

Example *(Add xyz macro to tools.dat)*

Suppose we wish to use the `\xyz` macro (with three arguments). What to do? The use could simply be `\loadxyz`, followed by the invocation `\xyz{...}{...}{...}`.

In order to make this possible, I have added to tools.dat

```
\tool\xyztool
\def\xyz#1#2#3{%the macro
...}
\endinput
```

and to blue.tex

```
\def\loadxyz{\let\xyztool=x
   \loadtool
   \let\xyztool\undefined
}
```

Confusing, isn't it?

Example *(Add an environment to tools.dat)*

Suppose we wish to make use of the environment

```
\begin⟨yourtoolname⟩
...
\end⟨yourtoolname⟩
```

In order to make this possible we have to do several things. First, add to tools.dat the following 'file'

```
\tool\⟨yourtoolname⟩tool%Opening
⟨themacros⟩
\endinput              %Closing
```

Second, include in blue.tex the environment opening and closing, next to the needed allocations

```
%Tool environment macro
\def\begin⟨yourtoolname⟩{\begingroup
   \let\end⟨yourtoolname⟩\endgroup
   \let\⟨yourtoolname⟩tool=x
   \loadtool
}
%storage allocation
\new⟨type⟩\⟨name⟩
%⟨type⟩ is  one of box count dimen fam insert language skip
%                   muskip read toks write
etc.
```

Fourth, verify the data-integrity. (See later.)

There are a few restrictions on the macros, as a consequence of the skipping technique used, i.e., via TEX's argument-reading mechanism.

- no outer defs, i.e, `\newtoks` and the like, should be declared in regular blue.tex
- no `\par`, use `\endgraf`, or `\csname par\endcsname`.

Remarks. The restriction with respect to your outer defs can be circumvented by

- defining all macros non-outer, or
- using `\csname newtoks\endcsname\`⟨*tag*⟩, or
- extending `\unouterdefs` by your outer defs, i.e., including `\newtoks`, and similars. When these are part of the format or tool, multiple use reallocates memory, and because TEX lacks a garbage collector this will cost extra memory locations
- sometimes a different macro name for blue.tex is needed, such as the index tool.

The `\tool` macro accepts any ⟨*yourtoolname*⟩

And what about double loading? At the moment it is printed in the log file, whenever a tool is loaded. Sometimes we wish to load a tool anew, for example the pascal environment. The index tool is only loaded once.[92] A load counter is maintained to control this.

[92] Very handy when after having proofed chapters with an index generated on the fly, we don't have to delete the `\loadindexmacros` tag in the chapters.

Check for data integrity After a tool has been added one can check whether the file can be scanned by TEX via making a table of contents as follows.

```
\input blue.tex
\contentstoolsorfmt{tools}
\bye
```

This script produces no pages of output (that is OK). The file contentstools will be created though.

Verbatim listing of a tool This problem is induced by my multi-file approach. Of course one can extract the tool by an editor et cetera as alternative.

Example *(Verbatim listing of macros of pascaltool)*

```
\input blue.tex \let\pascaltool=x
\long\def\tool#1{\ifx#1\undefined\bgroup\unouterdefs\ea\gobbletool
   \else\ea\toolverbatim\fi}
%\def\toolverbatim{\thisverbatim{\baselineskip12pt plus2pt minus1pt
%   \lineskip1pt plus1pt minus1pt}\beginverbatim}
\def\cgl;newcol{\endverbatim\endinput}
\input tools.dat \bye
```

Conventions for entries fmt.dat come down to

- update the contents at the beginning of the file tools.dat
- `\message{ ---`⟨...⟩`tool,` ⟨*date*⟩`,` ⟨*owner*⟩`--- }` in the beginning
- visual layout for print out
- additional comments which contain markup for line numbering
- addition of table of contents with cross-ref line numbers
- addition of history of changes

The layout is aimed at printing in two columns, and start a new column. The selective line numbering is controlled by the control sequences `\numvrb`, to start line numbering `\vrblin` = ⟨*number*⟩, to adjust line counter, and `\nonum` to switch off line numbering. Do remember that the escape char is the semi-colon.

Example *(Template for an entry in tools.dat)*

```
%end%%%%%%⟨lastlinepretioustool⟩%;newcol
%
%
%begin%%%%%%%%%%%%%%%%%%%⟨....⟩%;numvrb
\tool\⟨...⟩tool
\message{ ---⟨...⟩tool, ⟨date⟩,
                        ⟨(cr)owner⟩--- }
%⟨authorinformation⟩
%⟨thetoolproper⟩
\endinput                         %;nonum
%Contents
%⟨item⟩...................⟨pagenumber(s)⟩
%etc
%History of changes
%⟨date⟩ ⟨change⟩
%etc
%end%%%%%%%%%%%%%%%%%%%%%%⟨....⟩%!cgl;newcol
```

Bridge

This is available within the environment

`\beginbridge...\endbridge`.

the following has been copied from the `\bridgetool`.

```
%Typesetting bridge publications via (plain) TeX. Version May 92.
%The macros can als be used within LaTeX.
%Version: 1.1 March 1994 (basically unmodified since publication in 1990)
%Author: Kees van der Laan,Hunzeweg 57, 9893PB, Garnwerd (Gr)
%        The Netherlands. 05941-1525, cgl@rc.service.rug.nl
%Examples of use have been published in TUGboat, 11, 2, 265--276, and
%MAPS 91.2.
%The macros consist of 3 independent parts:
%- dec.tex   ((language) declarations, register used and control sequences)
%- bid.tex   (macros for bidding and layout of games)
%- play.tex  (macros for discussing the course of a play)
```

Example *(Bridge bidding, and play)*

The following has been copied from the `\bridgetool`. The 'course of play' has been printed in the chapter 'Tabular Material.'

```
%Examples: diagram, bidding, course of play.
%1. Diagram
$$\crdima{N/None}{\vtop{\hbox{Deal:}\hbox{demo}}}%
  {\hand{J74}{AJ}{QJT2}{Q874}}%N
  {\hand{K86}{T9542}{874}{T3}}%E
  {\hand{QT952}{Q83}{AK5}{A6}}%S
  {\hand{A3}{K76}{963}{KJ952}}%W
$$
%2. Bidding
$$\bbid
1\cl\alert& ? no& 1\sp&\think no\cr
       2\sp&  no& 4\sp&     a.p.\cr
\noalign{\vskip.5ex}%With the explanation
\alert\ means Alert, conventional bid\hidewidth\cr
? means explanation asked            \hidewidth\cr
\think means think pause             \hidewidth\cr
\ebid $$
%3. Course of play
\def\LFTINF{Puzzle} \def\RGTINF{\vtop{\hbox{6NT,}\hbox{by East}}}
\Ns={KQ76}\Es={T9}\Ss={8542}\Ws={AJ3}
\Nh={J98} \Eh={A2}\Sh={QT74}\Wh={K653}
\Nd={J942}\Ed={T5}\Sd={Q876}\Wd={AK3}
\Nc={65}\Ec={KJ9xxxx}\Sc={2}\Wc={AQT}
%
\showgame
%
*Problem* How must NS defend in order to guarantee 1 trick?
*Solution* Start with a \h\ lead in order to break communication.
N must discard \h s and S must discard \sp s.
\smallskip\noindent
\LEADS\bplay
h4 & hK & h8 & h2 & --& 1\LEADW\cr
cA  & c5 & cx & c2 & --& 2\cr
cQ  & c6 & cx & s2 & --& 3\cr
cT  & h9 & cK & s4 & --& 4\LEADE\cr
cJ  & s5 & s3 & s6 & --& 5\cr
```

```
c9  & s8 & h5 & s7 & --& 6\cr
cx  & d6 & sJ & d2 & --& 7\cr
\bintermezzo
\def\RGTINF{\vtop{\hbox to 0pt{NS squeezed on\hss}
             \hbox to 0pt{\cl\ continuation?\hss}
             \hbox to 7ex{\hss}}} %phantom for alignment
\Ec={{\ooalign{\hfil\raise.07ex%
    \hbox{x}\hfil\crcr\mathhexbox20D}}}
\showgame  \Ec={x}
\eintermezzo
cx & h7 & h6 & hJ & --& 8\cr
\omit et cetera \hidewidth \cr
\eplay
```

Crosswords

This is available within the environment

`\begincrosswords...\endcrosswords`.

For an example of use see the 'Tabular Material' chapter.

Tower of Hanoi

This is available within the environment

`\beginhanoi...\endhanoi`.

Within the environment the only useful command is `\hanoi{`⟨*number*⟩`}`. An example is provided in the 'Tabular Material' chapter.

Creation of an Index

This is different. `\loadindexmacros` should be provided at the beginning of the script. At the end supply `\sortindex`, followed by `\pasteupindex`. See the 'Creation of an Index' chapter for more details.

NTG logo

When `\loadntglogo` is invoked, the logo will be set in a vbox, `\ntglogobox`, for multiple use via for example `\copy\ntglogobox`.

Example *(Use NTG logo from tools.dat)*

```
\loadntglogo
\copy\ntglogobox
```

Nederlandstalige
TEX
Gebruikersgroep

Pascal texts

This is available within the environment

`\beginpascal...\endpascal`.

An example has been provided in the 'Text' chapter.

D: BLUes, BLUe's & Paradigms

The BLUe's format material and accompanying articles, which explain details and show examples, have been listed.

Originally this note was sent along with blue.tex to inform a user about background and related articles. *The TEXbook* is assumed to be omni-present. For other references consult lit.dat.

The files

BLUe's format has been organized as a 'kernel with modules.' The files are named as follows.

the format	**blue.tex**	the kernel
the databases	**fmt.dat**	the formats
	tools.dat	the macros and environments
	address.dat	the addresses
	lit.dat	the references
	pic.dat	the pictures
	obselete.dat	obsolete macros and formats.

A handy auxiliary is pfile—and the more general pgfile—to print blue.tex with appropriate line numbers in two-column layout. The general macro prompts for the file name. It is suited to print blue.tex, tools.dat, or fmt.dat, all with the appropriate line numbers.

This Publishing with TEX is the user guide.

No profit may be made on it. The articles are free for personal use. The files are shareware.[93]

When the results are used a proper credit—be it Knuth or the authors of gkpmac, or me for the composition and the fill-ins—is the least a happy publishing user can do.

Background

BLUe's Format—The best of both worlds	The umbrella article about blue.tex.
BLUe's Graphs—Simplicity, aha!	The gkppic macros are explained.
BLUe's Cross-Refs—A one-pass approach	One-pass cross-referencing with math.
BLUe's Bibliography—Selective loading	One-pass bibliography handling.
BLUe's Verbatim—The selection	Handling verbatims in TeX.
BLUe's Transparencies—To finish up with	The making of transparencies.
BLUe's Index—The principle and some more	The making of indexes in a one-pass job.
BLUe's Typesetting of PASCAL—An example	The formatting of PASCAL programs.
BLUe's Letters—With love	The formatting of letters.
BLUe's Reports—Good work	The formatting of reports.
BLUe's Format databases —Stay organized	The database approach explained.
Math into BLUes	The typesetting of math via TEX.
I: Mourning	
II: Sing your song	
Icons via lfonts	Common matrix icons via LATEX's fonts.
Table Diversions	The typesetting of tables via TEX.

93 When you are happy with it and wish to support maintenance, send me $ 25 to support my hardware, to maintain BLUe's format system. Of course you will be added to my register.blu database. Users from Russia can just send me their address, for the time being.

Sorting in BLUe	The basics for making a simple index.
FIFO and LIFO sing the BLUes	FIFO paradigm, fundamental.
Syntactic Sugar	Discusses various approaches to macros.
'Van tekst naar transparant'	About the life-cycle of documents, 1984.

Related material

Manmac BLUes —Or how to typeset a book via TEX	Is all about manmac, as-is.
AMS BLUes —Professionals at work	Details with AMS practice, `\lsams` for formatting references is inspired upon AMS.
TUGboat BLUes —How TEXies do it	TUGboat styles. ppt.tex emerged from this which is at the heart of blue.tex.

An overview is supplied in 'What is TEX and METAFONT all about?' NLUUG, 1993. The relation with SGML is in 'SGML(TEX,...),' Cork 1990. The work of Michael Spivak has been surveyed in Spivak's Œuvre.

For fun

Added are the following, which also show macro writing by example.

Typesetting bridge via TEX	MAPS 91.2
Typesetting crosswords via TEX	MAPS 92.2
Tower of Hanoi, revisited	MAPS 91.2

The Paradigm series

Paradigm: Plain's `\item` extended	MAPS 94.2
Paradigm: Headache?	MAPS 94.2
Paradigm: Two-part macros	MAPS 95.1
Paradigm: Parameterization I —Options	MAPS 95.1
Paradigm: Parameterization II —Place within context	MAPS 95.2
Paradigm: Naming conventions	MAPS 95.2
Paradigm: Alignment	MAPS 95.2
Paradigm: Eating and preserving	MAPS 95.2

Of course all the articles run up till 1995, with BLUe's format in plain TeX.

E: List of Examples

1 *Introduction*
1.1 *BLUe script* 3

2 *Text*
2.1 *Miniature* 11
2.2 *Flushed right paragraphs* 12
2.3 *Items* 12
2.4 *Quotations* 13
2.5 *Centering in display* 13
2.6 *Non-centered displays: one-line, multi-lines and multi-columns* 13
2.7 *Setting text apart* 14
2.8 *Syntax diagrams* 14
2.9 *Programs* 15
2.10 *Verbatims* 15
2.11 *Verbatims: escape char use* 15
2.12 *Verbatims: line numbering and file inclusion* 16
2.13 *Verbatims: line numbering* 16
2.14 *Verbatims: in line* 16
2.15 *Exercise and answer* 17
2.16 *Footnotes* 17

3 *Mathematics*
3.1 *Disparity math notation and markup notation* 18
3.2 *Automatic numbering and cross-referencing* 18
3.3 *Matrix icons* 19
3.4 *Compatible extension of eqalign with multiple alignment points* 19
3.5 *Macros for showing math markup and the result* 20
3.6 *Poor man's blackboard bold and some special symbols* 20
3.7 *A labeled formula in display* 21
3.8 *Formula hyphenation; shifting of lines* 21
3.9 *Alignment and centered labeling* 21
3.10 *Alignment and labels per line* 21
3.11 *Subscripts* 21
3.12 *Dots and the comma after* 21
3.13 *Summation with limits* 22
3.14 *Overlining* 22
3.15 *Square roots* 22
3.16 *Roman texts in math* 22
3.17 *Kerning; positive and negative* 22
3.18 *Empty formula and subscripting* 22
3.19 *Math operator* 22
3.20 *Colon markup; punctuation vs. operator* 22
3.21 *Context-dependent size* 22
3.22 *Vertical bars, The TEXbook 146, 147, ex18.21* 23
3.23 *Halves variety The TEXbook ex11.6* 23

3.24 *Under and overbraces* ... 23
3.25 *Diagonal dots, coercions, The TEXbook ex18.45* ... 23
3.26 *Undoing mathsurround space* ... 23
3.27 *All those Os* ... 24
3.28 *Set difference vs. cosets* ... 24
3.29 *The Cardano solution to third-order equation* ... 24
3.30 *Derivatives* ... 24
3.31 *Bessel equation* ... 24
3.32 *Primed summation symbols and split formula* ... 24
3.33 *Hypergeometric function* ... 25
3.34 *From The TEXbook 177, (p)matrix as formula part. proclaim is used too* ... 25
3.35 *Split equation and context-sized delimiters* ... 25
3.36 *Rhombus scheme* ... 25
3.37 *Commutative diagram, confer The TEXbook ex18.46* ... 26
3.38 *Partitioning* ... 26
3.39 *Continued fractions with alignment on =, and interruption* ... 26
3.40 *Matrix equation* ... 27
3.41 *Braces and matrices* ... 27

4 *Tabular Material*
4.1 *Just data, magic squares* ... 30
4.2 *Data plus header, The TEXbook 246* ... 30
4.3 *Data with row stubs, The TEXbook 232* ... 30
4.4 *Data, row stubs, header, caption and footer* ... 30
4.5 *Walter's spreadsheet, The TEXbook 244* ... 31
4.6 *Spreadsheet digression* ... 31
4.7 *Part of AAP's table, AAP 1989* ... 32
4.8 *AT&T table diversions, The TEXbook 247* ... 32
4.9 *Pittman's deterministic multiplication table* ... 33
4.10 *Young tableaux, irregular shape, Knuth 1973* ... 34
4.11 *Pascal triangle, a deterministic table* ... 34
4.12 *Tower of Hanoi, deterministic* ... 34
4.13 *Maze* ... 34
4.14 *Chart, The TEXbook ex22.14* ... 35
4.15 *Much alignment occurs in typesetting mathematics* ... 35
4.16 *Crosswords: puzzle, clues and solution* ... 35
4.17 *Chess boards* ... 36
4.18 *Optical illusion* ... 36
4.19 *Soto's kinetic art* ... 37
4.20 *Row and/or column spans, partial horizontal rules* ... 37
4.21 *Bridge pair match scheme, with special 'first' element* ... 38
4.22 *Alignment at decimal points* ... 38
4.23 *Nested table* ... 38
4.24 *Nested, and interrupted table with updating of memory* ... 38
4.25 *Fill-in forms* ... 39
4.26 *Time tables* ... 39
4.27 *No markup* ... 39
4.28 *Simultaneous row spans* ... 39
4.29 *Macroscopic markup vtable, based on valign* ... 40
4.30 *Simultaneous row and column spans* ... 40
4.31 *Gnonoms* ... 40
4.32 *ASCII table, The TEXbook 367* ... 41
4.33 *Markup for Appendix F 427, 428, 429, 430, 431, 432* ... 42

5 Pictures
5.1 Simple picture 44
5.2 Simple scheme 45
5.3 Manmanc's samplebox 45
5.4 Generalized samplebox macro 45
5.5 Atlas The TEXbook 389 46
5.6 Layout pages 46
5.7 Typesetting outlines, The TEXbook ex11.5 47
5.8 Simple picture markup 48
5.9 Entry for pic.dat 49
5.10 Flow chart loop: flowchartlooppic 50
5.11 Flow chart due to Furuta: furutapic 50
5.12 Salomon's overviewtexpic 51
5.13 SGML-TEX relation 51
5.14 Pie-chart: ntgpiepic 52

6 Coping with References
6.1 lit.dat entry 54
6.2 Formatting references your way 55
6.3 Search for references of 1973 in lit.dat 55

7 Creating an Index
7.1 Markup, commands and resulting index 56
7.2 IR markup 57
7.3 Spaces 57
7.4 Use of sorting keys 58

8 Customization
8.1 Page layout, and two-column settings 62
8.2 Default headline and footline code 63
8.3 Beginscript and endscript code 63
8.4 Head code 64
8.5 BLUe script into MAPS submission 64

B Formats
B.1 Verbatim listing of letter format 81
B.2 Template for an entry in fmt.dat 82
B.3 The soloists macro; block centering 83
B.4 A letter script; a letter to Don Knuth 84
B.5 Entry address.dat database 88
B.6 Search for addresses from RUSSIA in the database address.dat 89
B.7 A report script; the outer markup for this report 91
B.8 Addition of report format to the fmt.dat database 95
B.9 Transparencies script 96

C Tools
C.1 Add xyz macro to tools.dat 102
C.2 Add an environment to tools.dat 103
C.3 Verbatim listing of macros of pascaltool 104
C.4 Template for an entry in tools.dat 104
C.5 Bridge bidding, and play 105
C.6 Use NTG logo from tools.dat 106

F: Main Tags

A table of main tags of blue.tex has been included to provide a survey of what outer tags have been implemented for the regular format. More control sequences and symbols are available, especially for the lower level markup, if not for the undocumented features of manmac, and the wealth of plain TEX. The table also illustrates the naming convention adopted via a root name with prefixes and suffixes.

The header fields indicate, the prefixes pre-, post-, this-, every-, pasteup-, and the suffixes -font, -name and -box. The first column contains the various root names, with a marker to denote to what extent the minimal one-part macro has been implemented.

Main tags for regular scripts

Affix / Root		pre-	post-	this-	every-	pasteup-	-font	-name	-box
abstract	+	¬	¬	¬	¬	+	¬	+	+
acknowledgements	+	+	+	+	¬	+	¬	+	+
btable	▯	+	+	+	+	¬	¬	¬	+
center	+	¬	¬	¬	¬	¬	¬	¬	¬
contents	+	¬	¬	¬	¬	+	¬	+	+
(sub(sub))head	+	+	+	¬	¬	¬	+	¬	¬
keywords	+	¬	¬	¬	¬	+	¬	+	+
picture	¬	¬	¬	+	+	¬	¬	¬	¬
pictures	▯	¬	¬	¬	¬	¬	¬	¬	¬
quote	+	¬	¬	¬	¬	¬	¬	¬	¬
references	▯	+	+	+	+	+	¬	+	+
script	+	¬	¬	+	+	¬	¬	¬	¬
verbatim	¬	+	+	+	+	¬	¬	¬	¬

¬ not implemented
+ implemented
▯ implemented without processing on the fly

Remarks

- `\title`, `\subtitle`, `\author` and the like, are token variables and associated with the script
- `\pictures` and `\references` have been implemented in a straightforward way, because their use doesn't need processing on the fly
- In the Appendix Formats similar tables for `\transparencies` and `\report` have been included. The latter contains some more tags.

Your summary of markup tags Perhaps it is a good idea to add my Frequently Used Tags, FUTs—indeed a handful, as already indicated in the template in the 'Introduction'—for personalizing.[94]

Preliminary matter

```
\title{...}
\subtitle{...}
\keywords{...}
\abstract{...}
\contents{...}
\issue{...}
\acknowledgements{...}
\references{...}
\pictures{...}
```

Copy Proper

```
\beginscript
\endscript
\head{...}
\thissubhead{...}
\subhead{...}
\thissubsubhead{...}
(the \this⟨headkind⟩{...}  especially together with \runintrue)
\subsubhead{...}
```

Inner level

```
\thisverbatim{...}
\beginverbatim
⟨escapechar⟩endverbatim
(and in-line | ... |)
\begincenterverbatim
⟨escapechar⟩endcenterverbatim
\beginquote
\endquote
\begindisplay
\enddisplay
(with & and \cr)
\displaycenterline{...}
\btable
(with \data{...\cs...\rs...\cs...\rs...}
      \header{...}
      \footer{...}
      \rowstblst{...}
      \vruled{...} )
\ftn{...}
\bitem
\example{...}
\thispicture{...}
(with \unitlength=...)
\cs{⟨tagname⟩}
```

Back matter

```
\pasteupacknowledgements
\pasteupreferences
```

[94] I consider this very appropriate. At school the irony of cribs was when prepared carefully they made themselves superfluous. Transform these summary lists into your personal cribs.

G: History of Changes

An account of the history of BLUe's format has been provided. Milestones in the development have been listed.

In 1994 BLUe's format was introduced at CyrTUG and EuroTEX. The article was published in MAPS 94.1, and also in the EuroTEX proceedings (abbreviated). The fall article and a pre-release of blue.tex have been sent to the world, i.e., to the CTAN. At that tome only a default format and the transparencies format were available. It has been extended by report and letter format. BLUe's format design accounted for personalization, open-endedness, and a structural approach to coping with progress.

The major extension of 1995 was this user's guide, which entailed the `\report` format, and the inclusion of Knuth's letter and concert formats, *The TEXbook* Appendix E. Both were finished in January 1995. Also very important was adopting and keeping with a strong and clear organizational scheme, giving an answer the question how to cope with the multitude of formats, macros and tools out there. The files have been organized in databases. Not only a database for references, pictures, and addresses, no, also for tools and formats. What is needed will be loaded selectively from these, behind the scenes, and... by TEX. Neat!

Date	Change
Future	Updates to come
Dec 1994	list element tag in address.dat, lit.dat and pic.dat changed into `\lst` `\oldstyle` is introduced for the page numbers in the index etc.
	Font handling for transparencies has been cleaned up
	Addresses are stored in address.dat to cooperate with `\letter`
	Tools are stored in the database tools.dat
	Formats are stored in a database fmt.dat
	blue.tex has been extended to BLUe's format system `\report` and `\letter` have been designed `\nitem` uses `\prenum` and `\postnum` for embellishments `\create` in math cross-refs stores also `\prenum` and `\postnum` `\rowstblst`'s last argument is no longer an exception
	Partitioned matrix markup is short and elegant via `\btable`
	Space-saving variant of continued fraction looks much better now
	Dürer's magic square has a better `\dotboxit`.
	Bridge form details have been improved.

H: Obsolete database

To be filled in.

I: Index

In this index \end< *tag* >s have been excluded, only the \begin< *tag* >s are there.

<escapechar>endverbatim 15
\A4 62
\A5 62
AAP table 32
\add 58, 60
add a format 80, 95
add a picture 49
add a reference 54
add a tool 102
add an address 88
address.dat 6
\aitem 13
\Aitem 13
alignment at decimal point 38
alignment at number sign 38
\alt 15
AMS-TEX 6, 66
ASCII 6, 59
ATT table 33
\author 93, 100
\beginabstract 63
\beginbridge 105
\beginbtable 28
\begincenter 13
\begincenterblock 13, 83
\begincenterdisplay 13, 91
\begincenterverbatim 17
\beginchart 41
\begincontents 63
\begincrosswords 106
\begindemo 20, 91
\begindisplay 14
\beginhanoi 106
\beginhead 10
\beginkeywords 63
\beginlines 14
\beginmathdemo 18, 20
\beginpascal 15, 106
\beginpictures 44
\beginquote 13
\beginreferences 53
\beginscript 63, 91
\beginsyntax 15
\beginverbatim 15
Bessel equation 24
\bitem 13
blackboard bold 18, 20
BLU, BLUe 6
BLUe script 1, 6
BLUe script into MAPS 64
BLUe script, syntax 4
BLUe script, template 3
BLUe's format 1, 6
BLUe's format model 5
\bluepagepic 62
blue.tex 6
bordered table model 28
braces and matrices 27
btable, pitfalls 29
\btablecaption 28
\chapterhead 91
chart 35
chess 36
\circle 48
coercion macro 21
colon markup 23
commutative diagram 26
\conseqs 58
\consyms 58
continued fraction 26
cosets 24
cross-referencing 19, 53
crosswords 35
\crsref 19
\cs 28, 29
CTAN 6
\ctr 29
curly braces mania 10
customization 62
database, address.dat 83
database, fmt.dat 80
database, lit.dat 53
database, pic.dat 49
database, tools.dat 102
data with header 30
data with header, row stubs, caption and footer 31
data with row stubs 30
derivatives 24
design pages 46
diagonal dots 23
\disk 48
\displaycenterline 13, 91
\dotframed 29

\dotruled 29
dots and the comma after 22
dropped capital 11
empty formula 21
empty formula and subscripting 22
entry for address.dat 88
entry for fmt.dat 81
entry for lit.dat 54
entry for pic.dat 49
entry for tools.dat 102
\eqalign 18, 19
\everypicture 44
\everyreferences 53
fifo 88
\first 28, 38
\fll 29
flow chart 50
\flr 29
fmt.dat 6
\folio 64
font chart 42
fonts and size switching macros 17
\footer 28
\footline 63
footnote, automatic numbering 17
format, concert 82
format, letter 83
format, report 91
format, transparencies 96
formatting references 55
formula, automatic numbering 19
formula, cross-referencing 19
formula, hyphenation 21
formula, labeled 19, 21
\framed 29
framed boxes 45
\ftn 17
\gbox 45
\gkpbeginpicture 48
gkppic 5, 6
gnonoms 41
\halign 30
\head 64
\header 28
\headfont 64
heading, automatic numbering 10
heading, markup 10
\headline 63, 83
\hoffset 62
\hruled 29
\hsize 62
hypergeometric function 25
\hyphenchar 62
\iclIt 19
\icmat 19
\icuh 19
\icurt 19
ignored control sequences 58
\ignorewhitespace 64
IMHO 6
index, enrich 61
index, file 60
index.elm 60, 61
\indexfile 60
index.srt 60
IR 56
IR and spaces 57
IR, markup 57
IR, processes and files 60
IR, syntax 57
\irfile 60
\is 15
kerning; positive and negative 22
Knuth's markup 2, 9
LAMS-TEX 66
LATEX 6, 66
\leaders 36, 37
\letterto 88
\lettertoall 88
\line 48
line font 42, 47
list of database 49, 54
lit.dat 6
\loadindexmacros 56, 61, 106
\loadntglogo 106
logical multispan 29
\logms 29
\logo 100
look-and-feel 62
\lsams 55
maintenance transparencies 100
\makealllabels 89
\makebox 48
\makelabels 83, 89
\makexref 60
manmac 5, 6, 10
margin notes 17
markup, automatic, descriptive, functional, inner, Knuth's, logical, minimal, outer, recursive, structural, visual 6
Mathematical Writing 18
math operator 22
matrix equation 27
matrix icons 18
maze 35
miniature 7, 11
multi-line displays 14
multiple letters 88
multiplication table 33
\multiput 48
nested table 38
NFSS 7
\nitem 13
no markup 39
\nonframed 29
\nonruled 29
\normalchart 43
ntgpiepic 52

\numberstyle 59
\oldstyle 30, 42
\onecol 62
optical illusion 36
ordering 59
OTR 7
\ourreference 83
outline 47
\oval 48
overbraces 23
overlining 22
overviewtexpic 51
\pagewd 63
paragraph 11
paragraphs, flushed 12
partitioning 26
\partlogo 82
PASCAL texts 15
Pascal triangle 34
\pasteupindex 56, 60, 106
\pasteupreferences 53
PDF 7
pic.dat 7
\PiCTeX 7
picture environment 45, 47
\pictures 93, 100
\point 46
\postbtable 29
\posthead 64
\postnum 19
\postreferences 53
PostScript 7
\postverbatim 15
\prebtable 29
\prehead 64
\prenum 19
\prereferences 53
\preverbatim 15
processing on the fly 10
proofing single chapter 92
\put 48
quadratic spline 49
recursive markup 38
\ref 19
\references 93
report format, main tags 93
\rhombus 48
rhombus scheme 25
Roman numerals 64
roman texts in math 22
row and column spans 37, 40
row spans 40
\rowstblst 28
\rs 28
\ruled 29
Samelson principle 6
\samplebox 45
script, letter 84
script, report 91
script, transparencies 96
search of database 55, 89
separators 29
set difference 24
\setupnxtokens 60
SGML 7
sgmltexpic 51
size-switching macros 82
\soloists 82
\sortindex 56, 106
sorting keys 7
spreadsheet 31
\squine 49
\squineelem 49
\srtkeypairs 58
\subject 83
subscript 21
\subtitle 93, 100
\summary 91
Swanson 18
template 28
TEX 1
\thicklines 48
\thinlines 48
\thispicture 44
\thisreferences 53
\title 93, 100
tools.dat 7
tower of Hanoi 34
transparencies format, main tags 100
\tsaologo 82
TUGboat styles 68
\twocol 62
underbraces 23
\unitlength 44
\valign 40
\vector 48
verbatim text 15
verbatim, escape char 16
verbatim, file inclusion 16
verbatim, in-line 16
verbatim, line numbering 16
\voffset 62
\vruled 29
\vsize 62
\vtable 40
wiggling illusion 37
word ordering 59
\xcol 62
\xshift 45
\yields 20, 91
Young tableaux 34
\yourreference 83
\yshift 45

J: Table of Contents

1 *Introduction*
1.1 *The key to understanding Knuth's markup* 2
1.2 *BLUe's format* 3
The syntax 4
Variants 4
Extras 5
1.3 *The model* 5
1.4 *Notations and definitions* 6
1.5 *Markup conventions adopted* 7
1.6 *Copyrights BLUe's Format system* 8

2 *Text*
2.1 *Introduction* 9
2.2 *BLUe script* 9
2.3 *Headings* 10
Parameterization 10
2.4 *Paragraphs* 11
Parameterization 11
Special paragraphs 11
Customization 12
2.5 *Use of special text elements* 12
2.6 *Fonts and size switching* 17

3 *Mathematics*
3.1 *What's the problem, Doc?* 18
3.2 *The extras* 18
3.3 *Use* 20
Plain's display maths 21
A snapshot of examples 22

4 *Tabular Material*
4.1 *Bordered table* 28
Model 28
Markup tags 28
Customization 29
4.2 *Use of btable macro* 29
4.3 *Kaleidoscope* 34
Pondering about lists 39
Font charts 41
Why bother about font charts? 42

5 *Pictures*
5.1 *The roots* 44
5.2 *Using pictures* 44
5.3 *Creating pictures* 45
Stacking boxes 45
Samplebox diversions 45

Picture environment 47
The markup for the environment 48
Line thickness 48
(multi)put 48
Picture objects 48
Curves 49
5.4 Adding a picture to the database pic.dat 49
Check for data integrity 49
5.5 Examples 50
6 Coping with References
6.1 Using references 53
Cross-referencing 53
Robustness 53
6.2 Adding a reference to the database lit.dat 54
Check for data integrity 55
6.3 Customization 55
How to provide your way of formatting? 55
6.4 Searching the lit.dat database by pattern 55
7 Creating an Index
7.1 Use 56
7.2 Markup of Index Reminders 57
Syntax 57
Markup 57
Spaces 57
7.3 Special tokens 58
Tokens to be ignored 58
Sorting keys 58
7.4 Ordering 59
7.5 Typesetting the index 59
7.6 Customization 60
Adding tokens 60
Modifying ordering 60
The process and files involved 60
Enriching the index 61
Typesetting the enriched file 61
8 Customization
8.1 Changing the look-and-feel 62
Adapting the page size 62
The header and footer 63
Title part 63
Section beginnings 64
Raster layout and TEX 64
8.2 BLUe script into MAPS submission 64
8.3 Other languages 65
9 Afterthoughts
9.1 AMS-(La)TEX 66
9.2 LAMS-TEX 66
9.3 LaTEX 67
9.4 TUGboat styles 68
9.5 The future 68
A Table of Contents 68
An Index 68
Cross-referencing 68
The bad news 69
9.6 Statistics about processing PWT 69

A Answers to the exercises

B Formats

B.1 Use of formats from fmt.dat database 80
Adding a format 80
Check for data integrity 81
Verbatim listing of a format 81
Conventions for entries fmt.dat 81
B.2 Concert format 82
Centering a block 83
B.3 Letter format 83
Use of the letter format 84
Syntax entries address database 88
Check for data integrity 88
Multiple letters 88
And what about a letter to the complete address database? 88
Address labels 89
Searching the address.dat database by pattern 89
Customization 89
Your summary of main markup tags for letters 89
B.4 Report format 91
Template 91
Proofing a single chapter 92
Main tags for report format 93
Your summary of main markup tags for reports 93
Addition of a format to BLUe's format system 95
B.5 Transparencies format 96
Template 96
Use your logo 100
Maintenance 100
Main tags for transparencies 100
Your summary of markup tags for transparencies 101

C Tools

C.1 Use 102
Adding a tool 102
And what about double loading? 103
Check for data integrity 104
Verbatim listing of a tool 104
Conventions for entries fmt.dat 104
C.2 Bridge 105
C.3 Crosswords 106
C.4 Tower of Hanoi 106
C.5 Creation of an Index 106
C.6 NTG logo 106
C.7 Pascal texts 106

D BLUes, BLUe's & Paradigms

D.1 The files 107
D.2 Background 107
D.3 Related material 108
D.4 For fun 108
D.5 The Paradigm series 108

E List of Examples

F Main Tags

Your summary of markup tags 113

G History of Changes

H Obsolete database

I Index

*N*ederlandstalige
*T*eX
*G*ebruikersgroep

is the Dutch-language-oriented TeX Users Group, founded in 1988. The activities comprise two meetings per year, with the wealthy MAPS—Minutes and APpendiceS—sent out before each meeting. The activities are biased by sharing experience in using (LA)TeX, METAFONT, literate programming, SGML, Adobe's PostScript and PDF, next to multi-media, in short, sharing knowledge about Electronic Publishing, biased by (LA)TeX. The intricacies of the World-Wide-Web in relation to using (LA)TeX is also explored. Less than 5% of our energy was devoted to organizational matters in the past. Apart from the regular MAPS we enjoy MAPS specials. Up till 1994 NTG's PR set and David Salomon's courseware—Advanced TeX: Insights and Hindsights—have appeared in this series. The 1994 special has the title 100 FAQ: Everything you will ever have to know about (LA)TeX.

Important activities, listed in historical order, have been

1988	NTG's discussion list and fileserver on the air
1989	NTG days, various courses have been offered since
1990	SGML-TeX conference
1992	Belgian commissioner for Flemish (LA)TeXies
1993	Goddijn's Bulletin Board
1994	4AllTeX CD-ROM with turnkey-TeX system.

Close contacts have been maintained over the years with the TUG—TeX Users Group—and the other LUGs—Language-oriented TeX Users Groups. One can subscribe for a joint membership of NTG and TUG.

The regular subscription fee for an individual NTG membership is in the order of fl.90 per year. Students are granted a discount. An institutional membership is possible.

Enquiries, casu quo a registration form, can be obtained from NTG's secretary Postbus 394, 1740 AJ Schagen, ntg@nic.surfnet.nl.

Thank You

`\bye`