Indexing in TEX with AnyTEX

Kees van der Laan

Hunzeweg 57
9893 PB Garnwerd
The Netherlands
cgl@rc.service.rug.nl

Abstract

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.

1 Introduction

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 – also called mini-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.[1]

This paper is essentially a chapter from the user's guide 'Publishing with TEX,' which comes with BLUe's Format system.

2 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.

1. It is said that the automatic generation of an index is a feature of the Literate Programming tools. For LP with TEX as such, as for example Gurari's ProTEX, this on-the-fly indexing within TEX can be used.

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
\controlsequence1
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.

Question 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?[2]

Answer 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.'

2. Courtesy Erik Frambach.

3 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.

3.1 Syntax

Knuth's IRs obey the following syntax. IR, syntax

<word(s)>␣!<digit>␣<page number>.

The digits 0, 1, 2, or 3 denote the types: words, verbatim words, control sequences, and syntactic quantities. 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.[3]

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

* |...| 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.

3.2 Markup

The markup for IRs is near to natural. Precede the entry by a circumflex, or a double one in case of a silent[4] 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
```

3. 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.
4. Silent IRs mean that these will appear only in the index, not on the page.

3.3 Spaces

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 \␣.

Question What to do when part of a title should reappear in the index?
Answer 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.
Example Spaces

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.

```
^{\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.
```

3.4 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.[5] 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.

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

3.5 Tokens to be ignored

In practice I needed things like `\tt` as part of the IR, which must be neglected while sorting.[6] 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{\‘\’\"\^\~}
```

3.6 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.

```
\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.

Question What to do when ‘to’ is part of the sorting key?
Answer Add an extra level of braces.

4 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

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

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.[7]

Furthermore, a user can select the so-called word ordering,[8] 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		

5 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`.

6 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.

---

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

6.1 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.[9]

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.[10]

6.2 Modifying ordering

A general way is to 'copy' the ordering table and to modify it.[11]

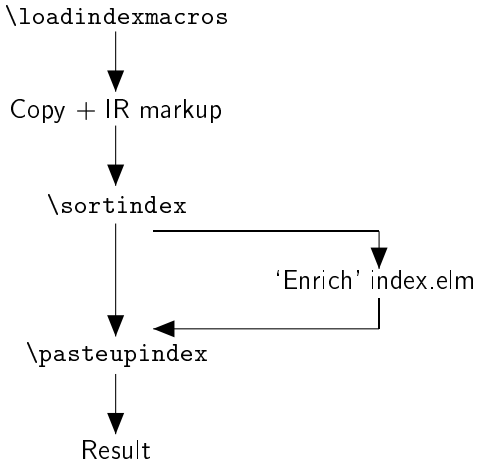
And what about a macro to add to the table? This can be done easily, and superficially looks convenient for an innocent user. At the moment I don't trust the macros to be worthwhile for an innocent user, unless a very modest index has to be made. And this completes the circle: different ordering is not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

6.3 The process and files involved

Like in manmac, `blue.tex` stores the raw IRs in the file index. The file index[12] is read and stored in an array for internal sorting. After sorting, the number of entries is reduced,[13] and the result is written to the file `index.srt`. Then, `index.srt` is transformed into the file index.elm.[14] The result is typeset via `\pasteupindex`. Schematically it comes down to the following.

---

9. It is simpler to add those control sequences to `index.elm`.
10. `\noexpand` is used instead of `\string`.
11. 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.'
12. Default index is the value of the toks variable `\irfile`, which is used in `\sortindex`.
13. Those which differ by page number are collected in one entry.
14. 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.[15]

6.4 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.

6.5 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`.

7 Extras

Ubdoubtedly people favor their own subset of TEX, or more likely LATEX. The good news is you don't have to use BLUe's format system. I gathered the sorting and indexing stuff as an indpendent self-contained set in the file `plainindex.tpl`.

The bad news is that up till now I did not do much about preventing name clashes.

15. 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.

7.1 TEXnical details

The details with respect to indexing have been treated in 'BLUe's Indexes,' and the sorting aspects have been treated in 'Sorting in BLUe,' both available from the CTAN, in the directory `/pub/archive/info/pwt/<filename>`