

---

## Glisterings: Index headers; Numerations; Real number comparison

Peter Wilson

---

Gaul as a whole is divided into three parts.

---

*De Bello Gallico*, JULIUS CAESAR

The aim of this column is to provide odd hints or small pieces of code that might help in solving a problem or two while hopefully not making things worse through any errors of mine.

*La dernière chose qu'on trouve faisant un ouvrage, est de savoir celle qu'il faut mettre la première.*

The last thing one knows in constructing a work is what to put first.

---

*Pensées*, BLAISE PASCAL

### 1 Index headers

In the manual for the memoir class the headers for the index included the first and last main index entries on the page in question. The technique I used was fairly simple and can be applied to any document whose index is generated via the MakeIndex program. It consists of defining a macro in the document's preamble (or a package) and a simple MakeIndex style file [4, 8, 16].

Below is an example of a document file, where the new macro is `\idxmark` and the `fancyhdr` package [15] is used for specifying the page headers and footers.

`\idxmark{<entry>}` prints `<entry>` and also sets both the marks to `<entry>`, where `\leftmark` resolves to the first mark on the page and `\rightmark` to the last mark on the page.

```
\documentclass[...twoside]{...}
\usepackage{makeidx,fancyhdr}
\newcommand*\idxmark[1]{#1\markboth{#1}{#1}}
\pagestyle{fancy}
%% set up general fancy headers/footers
\makeindex
\begin{document}
... % many \index{...} ...
\clearpage
%% fancy headers/footers for index pages
\fancyhead{}
\fancyhead[LE,R0]{%
  \rightmark\space---\space\leftmark}
\fancyfoot{}
\fancyfoot[C]{\thepage}
\printindex
\end{document}
```

Peter Wilson

The index is typically set in two columns and L<sup>A</sup>T<sub>E</sub>X's marking system did not always function well in this case; in the past, including the `fixltx2e` package was needed to fix that, but nowadays it is included by default.

In the example the general header/footer styles are set in the preamble using the `fancyhdr` package facilities. The (special) headers and footers for the index are set just before the index commences. When looking for something in a book I flick it half closed only seeing the outer portion of each page. Consequently, in this case I have put the first and last index entries on the page at the outer of the headers where they are easy to see and the page number, which is not particularly important when looking through the index, centered at the foot of the page.

Here are the essentials of the MakeIndex style file for the application. For each main entry, `<entry>` in the input `.idx` file it outputs `\idxmark{<entry>}` in the `.ind` file which is used by `pdflatex` to print the index.

```
% MakeIndex style file flindex.ist
% output main entry as <entry> as:
%   \item \idxmark{<entry>}
item_0 "\n\item \idxmark{"
delim_0 "}, "
% not forgetting subitem
\item_X1 "} \n \subitem "
```

The memoir manual, `memman.pdf`, has an index that is over 40 pages long with each alphabetic section headed by the letter (A, B, C, etc.). The `hyperref` package is used to enable PDF bookmarks and Lars Madsen developed code so that the index letter subheads would appear in the bookmark listing. Here is the skeleton of how this was done, where the `\doidxbookmark{<head>}` macro does the work. The `<head>` argument is an index subhead and the macro adds it to the list of bookmarks at one level below the Index entry. The default MakeIndex generated subhead for entries that do not commence with an alphanumeric is 'Symbols', but I felt that 'Alphabetics' was a better subhead. `\doidxbookmark` prints the appropriate subhead centered in a bold font and adds it to the bookmark list.

```
...
\usepackage{ifpdf}
\ifpdf
  \usepackage[pdftex,
    plainpages=false,
    pdfpagelabels,
    bookmarksnumbered,
    colorlinks,
    ocolorlinks,
  ]{hyperref}
\else
```

```

\usepackage[plainpages=false,
            pdfpagelabels,
            bookmarksnumbered,
            colorlinks,
            ]{hyperref}
\fi
\makeatletter
\newcommand*\doidxbookmark}[1]{%
  \def\@tempa{Symbols}\def\@tempb{#1}%
  \centering\bfseries \ifx\@tempa\@tempb
    Alphabetic
    \phantomsection%
    \belowpdfbookmark{Alphabetic}%
    {Alphabetic-idx}%
  \else
    #1%
    \phantomsection%
    \belowpdfbookmark{#1}{#1-idx}%
  \fi
  \vskip\baselineskip\par}
\makeatother
...
\begin{document}
...
\clearpage
\pdfbookmark{Index}{Index}
\phantomsection
\printindex
\end{document}

```

The MakeIndex style file must be written so that it wraps the `\doidxbookmark` around the subheads, like this:

```

% MakeIndex style file flindex.ist
% output main entry ...
% Wrap and uppercase head letters
headings_flag 1
heading_prefix "\\doidxbookmark{"
heading_suffix "}"

```

And here is the sequence of commands to generate the indexed file.tex document

```

> pdflatex file
> makeindex -s flindex.ist file
> pdflatex file

```

As yet a child, nor yet a fool to fame,  
I lisped in numbers, for the numbers came.

---

*An Epistle to Dr Arbuthnot,*  
ALEXANDER POPE

## 2 Numerations

### 2.1 Sorted lists

Reza wrote to ctt [12] saying that he had developed code that would sort the items in a description en-

vironment and asked if there was a way to enumerate the entries.

I had no idea that you could sort items within L<sup>A</sup>T<sub>E</sub>X but it appears that Nicola Talbot's incredible `datatool` package [14] enables you to do that, and much more.

Here is Reza's original code, to which I have added some comments to try and indicate what is happening and changed the environment name to distinguish it from other later code:

```

%%%% Reza's code
\usepackage{datatool}
\newcommand{\sortitem}[2]{%
  % start a new row in the db
  \DTLnewrow{list}%
  % add key/value to row
  \DTLnewdbentry{list}{label}{#1}%
  % add another key/value to the row
  \DTLnewdbentry{list}{description}{#2}%
}
\newenvironment{sorteddesc}{%
  % use or create a db called 'list'
  \DTLifdbexists{list}
  {\DTLcleardb{list}}{\DTLnewdb{list}}%
}{%
  % at the end of the environment sort the
  % db in ascending order of the label
  \DTLsort{label}{list}%
  % start a description environment
  \begin{description}%
    % iterate through the db,
    % picking out the keys/values
    \DTLforeach*{list}%
      {\theLabel=label,\theDesc=description}{%
        \item[\theLabel]\theDesc
      }%
  \end{description}%
}

```

As an example,

```

\begin{sorteddesc}
\sortitem{zz}{description of zz}
\sortitem{mm}{description of mm}
\sortitem{aa}{description of aa}
\end{sorteddesc}

```

results in:

```

aa description of aa
mm description of mm
zz description of zz

```

Reza wanted to know if there was a way to enumerate the entries. That is, so the output would look like:

1. aa description of aa
2. mm description of mm
3. zz description of zz

Christian Anderson [1] responded with the following based on using the `enumerate` environment (I have changed the name of the sorting environment to distinguish it from the other proposals):

```
\newenvironment{sortedenum}{%
  \DTLifdbexists{list}%
  {\DTLcleardb{list}}{\DTLnewdb{list}}%
}{%
  \DTLsort{label}{list}%
  \begin{enumerate}%
    \DTLforeach*{list}%
      {\theLabel=label,\theDesc=description}{%
        \item \theLabel\ \theDesc% original
        % bolds the descriptive label
      }%
    \item \textbf{\theLabel}\ \theDesc%
  }%
\end{enumerate}%
}
```

As an example of Christian's proposal

```
\begin{sortedenum}
\sortitem{zz}{description of zz}
\sortitem{mm}{description of mm}
\sortitem{aa}{description of aa}
\end{sortedenum}
```

results in:

1. aa description of aa
2. mm description of mm
3. zz description of zz

It was unclear as to how Raza wanted the combination of number and label to be typeset. Christian indicated how the label could be typeset in a bold font, but the number would still be set in the normal font.

Alan Munn [9] also responded as follows, taking advantage of the `\DTLcurrentindex` macro from `datatool` (again I have changed the name of the sorting environment).

```
\newenvironment{sorteditemdesc}{%
  \DTLifdbexists{list}%
  {\DTLcleardb{list}}{\DTLnewdb{list}}%
}{%
  \DTLsort{label}{list}%
  \begin{description}%
    \DTLforeach*{list}%
      {\theLabel=label,\theDesc=description}{%
        \item[{\normalfont
          \DTLcurrentindex.\ }
          \theLabel]\theDesc}%
      }%
  \end{description}%
}
```

As an example of Alan's proposal

```
\begin{sorteditemdesc}
\sortitem{zz}{description of zz}
\sortitem{mm}{description of mm}
```

```
\sortitem{aa}{description of aa}
\end{sorteditemdesc}
```

results in:

1. **aa** description of aa
2. **mm** description of mm
3. **zz** description of zz

Alan also indicated how the number could be set in bold to match the label. By suitable application of `\normalfont` both could instead be set in the normal font. In this sense Alan's solution is slightly more general than Christian's.

## 2.2 Autotab

Jeremy wrote to ctt [7]:

*I want to create a table and I want the first column of the table to have incrementing numbers ... I would like to avoid having to manually enter the numbers in the first column. Is there a way to do this automatically?*

Alan Munn responded with [10]:

```
%\usepackage{booktabs}
%\usepackage{array}
\newcounter{rownum}
\newcolumnntype{N}{%
  >{\stepcounter{rownum}\therownum.\ }1}
\newcommand*{\resetrownum}{%
  \setcounter{rownum}{0}}
\begin{tabular}{N11} \toprule
\multicolumn{1}{>{\resetrownum}1}{I}
& A & B \\ \midrule
& blah & blah \\
& foo & foo \\ \bottomrule
\end{tabular}
```

I	A	B
1.	blah	blah
2.	foo	foo

In the same thread Heiko Oberdiek noted that the `\resetrownum` could be taken out of the `tabular`:

```
\resetrownum
\begin{tabular}{N11} \toprule
\multicolumn{1}{1}{I}& A & B \\ \midrule
& blah & blah \\
& foo & foo \\ \bottomrule
\end{tabular}
```

In a later thread on a similar but extended topic Romildo posed [13],

*I want to typeset an enumeration list in a tabular format, building a table with automatic numbered items and sub-items in different rows. [An example layout followed.]*

Jean-François Burnol [3] responded with:

```

\newcounter{bitmi}
\renewcommand{\thebitmi}{%
\arabic{bitmi}}
\newcounter{bitmii}[bitmi]
\renewcommand{\thebitmii}{%
\thebitmi.\arabic{bitmii}}
\newcounter{bitmiii}[bitmii]
\renewcommand{\thebitmiii}{%
\thebitmii.\arabic{bitmiii}}
\newcommand{\bitm}{\stepcounter{bitmi}
\hbox to 1.5em{\thebitmi.\hfil}}
\newcommand{\bsubitm}{\stepcounter{bitmii}
\hbox to 1.5em{
\hbox to 2.5em{\thebitmii\hfil}}
\newcommand{\bsubsubitm}{%
\stepcounter{bitmiii}
\hbox to 4em{
\hbox to 3.5em{\thebitmiii\hfil}}

\begin{tabular}{lllll}
Subject & & Class & & Total & & Notes & \\\
\bitm First topic & 2 & & 2 & & a, b & \\\
\bitm Second topic & 12 & & 14 & & & \\\
\bsubitm Aaaa & & & & & b, c, d & \\\
\bsubsubitm M1 & & & & & a & \\\
\bsubsubitm M2 & & & & & b, e, f & \\\
\bsubitm Bbbb & & & & & a, b & \\\
\bitm Third topic & \ldots & & & & & \\\
\end{tabular}

```

Subject	Class	Total	Notes
1. First topic	2	2	a, b
2. Second topic	12	14	
2.1 Aaaa			b, c, d
2.1.1 M1			a
2.1.2 M2			b, e, f
2.2 Bbbb			a, b
3. Third topic	...		

Just a little later Peter Flynn proposed [5]:

```

\usepackage{array}
\newcounter{topic}
\renewcommand{\thetopic}{\arabic{topic}}
\newcounter{topici}[topic]
\renewcommand{\thetopici}{%
\thetopic.\arabic{topici}}
\newcounter{topicii}[topici]
\renewcommand{\thetopicii}{%
\thetopici.\arabic{topicii}}
\newcommand{\topic}[1]{%
\stepcounter{topic}%
\vrule height1.2em width0pt\thetopic. &
\multicolumn{3}{l}{\#1}}
\newcommand{\subtopic}[1]{%
\stepcounter{topici}%
\vrule height1em width0pt & \thetopici &
\multicolumn{2}{l}{\#1}}
\newcommand{\subsubtopic}[1]{%

```

```

\stepcounter{topicii} & &
\thetopicii\#1}

\begin{tabular}{rrrrl<{\quad}r<{\enspace}l}
\multicolumn{4}{l}{\textbf{Subject}} & &
\multicolumn{1}{r}{\textbf{Class}} & &
\multicolumn{1}{r}{\textbf{Total}} & &
\textbf{Notes} \\ \[2pt]
\topic{First topic} & 2 & 2 & a, b & \\\
\topic{Second topic} & 12 & 14 & & \\\
\subtopic{Aaaa} & & & b, c, d & \\\
\subsubtopic{M1} & & & a & \\\
\subsubtopic{M2} & & & b, e, f & \\\
\subtopic{Bbbb} & & & a, b & \\\
\topic{Third topic} & \ldots & & & \\\
\end{tabular}

```

Subject	Class	Total	Notes
1. First topic	2	2	a, b
2. Second topic	12	14	
2.1 Aaaa			b, c, d
2.1.1 M1			a
2.1.2 M2			b, e, f
2.2 Bbbb			a, b
3. Third topic	...		

Jean-François's approach to me is the simpler and easier of the two as he designed it so that all the main entries go into the first column and internally uses empty `\hboxes` to indent the several (sub) item levels. Peter's approach is more complex in that it involves several `\multicolumns` to control the (sub) topic indentations; he also uses zero-width vertical rules to enable different vertical spacing between the topic levels. My feeling is that the combination of Jean-François's horizontal positioning and Peter's vertical adjustments might be closest to an optimum solution to Romildo's needs.

Tenants of life's middle state,  
Securely placed between the small and great.

*Tirocinium*, WILLIAM COWPER

### 3 Real number comparison

On `ctt` Pluto noted that `\ifnum` could be used to compare integer values and wondered if there was a similar method for comparing real numbers [11]. This led to a spirited discussion involving several people and over 50 postings at my last count. The two that struck me the most were from Donald Arseneau and 'GL'.

For demonstration purposes assume that the following are defined:

```
\def\fourfive{4.5}
```

```
\def\fivefive{5.5}
\def\sixseven{6.7}
\def\threetwo{3.2}
```

Donald gave concise and elegant code,<sup>1</sup> shown later, that could be used like this:

```
\ifnum <condition> \then...\else...\fi
as in:
\ifnum 4.5 > 5.5
  \then 4.5 is larger than 5.5
  \else 4.5 is not larger than 5.5\fi. \
\ifnum \sixseven > \threetwo
  \then \sixseven\ is larger than \threetwo
  \else \sixseven\ is not larger than
    \threetwo \fi.
```

```
4.5 is not larger than 5.5.
6.7 is larger than 3.2.
```

Here is Donald Arseneau's code [2]:

```
\let\then\iffalse
\def\gobblejunk#1\delimiter{}
\def\ifnum#1\then{\ifdim
  \ptlt\ptgt\pteq #1pt\gobblejunk<=>\delimiter}
\def\ptlt#1<{\#1pt<}
\def\ptgt#1>{\#1pt>}
\def\pteq#1={\#1pt=}
```

GL also provided code, shown later, that could be used like this:

```
\unless\Realnums\ifnum <condition>
  \Then ... \else ... \fi
as in:
\unless\ifnum\Realnums\fourfive>\fivefive
  \Then \fourfive\ is not larger than \fivefive
  \else \fourfive\ is larger than
    \fivefive\fi. \
\unless\ifnum\Realnums 6.7 > 3.2
  \Then 6.7 is not larger than 3.2
  \else 6.7 is larger than 3.2\fi.
```

```
4.5 is not larger than 5.5.
6.7 is larger than 3.2.
```

In GL's original posting [6], which was in response to Donald, he reused, modified, and extended Donald's code. I have renamed some of the macros in GL's code so that the two sets are distinct, which means that both styles can be used in the same document, as I have done here.

```
\def\gobblejunk#1\delimiter{}
\def\Realnums#1\Then{\dimexpr
  \Ptlt\Ptgt\Pteq #1pt\gobblejunk<=>\delimiter}
\def\Ptlt#1<{\#1pt<\dimexpr}
\def\Ptgt#1>{\#1pt>\dimexpr}
\def\Pteq#1={\#1pt=\dimexpr}
```

<sup>1</sup> Which I cannot interpret for you.

The `\unless` macro is defined in  $\epsilon$ -TeX, which all recent L<sup>A</sup>T<sub>E</sub>X systems automatically utilise. Personally I find the `\unless` construct hard to get my mind around; it's the reverse of the traditional `\if...`

## References

- [1] Christian Andersen. Re: Sorted items in description environment. `comp.text.tex`, 10 October 2010.
- [2] Donald Arseneau. Re: `\ifnum` for real numbers. `comp.text.tex`, 16 October 2010.
- [3] Jean-François Burnol. Re: Tabular with enumerated items and sub-items in different rows. `comp.text.tex`, 27 April 2011.
- [4] Pehong Chen and Michael A. Harrison. Index preparation and processing. *Software Practice and Experience*, 19(8):897–915, September 1988. <http://mirror.ctan.org/indexing/makeindex/paper>.
- [5] Peter Flynn. Re: Tabular with enumerated items and sub-items in different rows. `comp.text.tex`, 27 April 2011.
- [6] GL. Re: `\ifnum` for real numbers. `comp.text.tex`, 16 October 2010.
- [7] Jeremy. Table with auto-incrementing column. `comp.text.tex`, 11 April 2011.
- [8] Frank Mittelbach and Michel Goossens. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison Wesley, second edition, 2004. ISBN 0-201-36299-6.
- [9] Alan Munn. Re: Sorted items in description environment. `comp.text.tex`, 10 October 2010.
- [10] Alan Munn. Re: Table with auto-incrementing column. `comp.text.tex`, 11 April 2011.
- [11] Pluto. `\ifnum` for real numbers. `comp.text.tex`, 10 October 2010.
- [12] Reza. Sorted items in description environment. `comp.text.tex`, 10 October 2010.
- [13] Romildo. Tabular with enumerated items and sub-items in different rows. `comp.text.tex`, 27 April 2011.
- [14] Nicola L.C. Talbot. `datatool`: Databases and data manipulation, 2016. <http://ctan.org/pkg/datatool>.
- [15] Piet van Oostrum. Page layout in L<sup>A</sup>T<sub>E</sub>X, 2016. <http://ctan.org/pkg/fancyhdr>.
- [16] Peter Wilson. The memoir class for configurable typesetting, 2016. <http://ctan.org/pkg/memoir>.

◇ Peter Wilson  
12 Sovereign Close  
Kenilworth, CV8 1SQ UK  
herries dot press (at)  
earthlink dot net