

Guide to 2016 Extensions in \LaTeX Typesetting of University of Utah Dissertations and Theses

Nelson H. F. Beebe

25 March 2016

Contents

1	Introduction and background	1
2	The \LaTeX package problem	3
3	The 2016 thesis file structure	4
4	Colored section headings	6
5	Running page headers	8
6	Other useful packages	9
7	Final formatting tweaks	11
8	Creating a subject index	13
9	What is unlikely to work	20

1 Introduction and background

IT TAKES A LOT OF ENERGY TO FIRE UP A STEAM LOCOMOTIVE,
BUT ONCE IT IS ROLLING, IT IS HARD TO STOP.
WRITING A DISSERTATION IS LIKE THAT.

The University of Utah has long had \LaTeX support for writing of theses and dissertations, with a collection of files available at the equivalent Internet sites

<ftp://ftp.math.utah.edu/pub/uuthesis>
<http://www.math.utah.edu/pub/uuthesis>

However, the increasing number of packages available in \TeX distributions on modern platforms, combined with changes in formatting styles mandated by the University of Utah Thesis Office, have made it difficult for student authors to conform to those styles.

Too much time has been wasted by thesis editors, and by students fiddling with \LaTeX formatting to try to achieve conformance and thesis acceptance. The 2016 revisions of the \LaTeX support have been coordinated closely with the Thesis Office to try to eliminate that waste and frustration in formatting.

Much of the formatting problems stem from overly strict specifications that have nothing to do with content, but arise because the Thesis Office does not want to have different formatting rules for documents produced with word-processing software, and with \LaTeX .

Those two classes of document formatters have quite different views of document production. In particular, word processors generally set text on a grid of lines, whereas the \TeX typesetting engine that underlies the \LaTeX document production system has no concept of fixed grids, and instead works in a model of *boxes and glue*.

In \TeX , words go into boxes with letters separated as specified by the font designer, then those boxes are assembled into lines when the text is ready to be shipped as part of a completed page to the output file. The boxes within each line are separated by *glue* (think ‘rubber cement’), which is essentially space that can be shrunk or stretched by user-controllable amounts, although those amounts are most commonly defaults that are set by the font designer. In addition, in complex documents, and especially those containing mathematical text, which is inherently *two-dimensional*, the objects to be assembled on the page do not have predictable vertical sizes, and those sizes are rarely constrainable to a fixed grid.

In normal documents typeset by \TeX , there are usually multiple places on the page where vertical glue allows sufficient flexibility for \TeX to set the page with a balanced and uniform appearance. Such glue sometimes occurs between lines (though it is likely to be tiny), with larger amounts specified between paragraphs, and above and below nonstandard material, such as headings, mathematical displays, lists, quotations, figures, tables, and so on.

The Thesis Office now mandates no additional vertical space, and zero glue, between paragraphs, but does allow glue to surround most of the other objects on the page, notably, mathematical displays and lists. As a result, thesis pages are typeset with the \TeX `\raggedbottom` command in effect, which essentially supplies an infinite amount of glue at the end of each completed page, so that \TeX does not complain about underfull `\vbox` conditions on almost every page.

2 The \LaTeX package problem

WESLEY OPENED THE PACKAGE AND LAID A BROWN LEATHER
LUNCH BOX ON THE TABLE. "MIGHT BE A COUPLE OF BOOKS,
OR DRAWING TOOLS OR MOST ANYTHING THAT'S NEAT AND GENTEEL.
YOU SEE, IT OPENS THIS WAY."

GENE STRATTON PORTER, *A Girl of The Limberlost* (1909).

The \LaTeX document formatting system layers a huge structure on top of the \TeX engine to allow documents to be written with declarations of what various pieces of text *are*, rather than with what they *look like* on a typeset page. Thus, a \LaTeX user can treat all documents as similar: they only have to be described by markup that identifies *front matter*, *parts*, *chapters*, *sections*, *subsections*, ..., *equations*, *figures*, *tables*, and so on. The visual appearance of those objects is then relegated to the internals of a *document class* file, and possibly further modified by the user's inclusion of an arbitrary number of \LaTeX packages with `\usepackage` commands.

The design of the original class file for University of Utah theses and dissertations expected \LaTeX input to look something like this:

```
\documentclass {uthesis2e}
\title {...}
\author {...}
\date {...}
... front matter definitions ...
\begin {document}
... front matter output ...
\tableofcontents
\listoffigures
\listoftables
\include {chap1}
\include {chap2}
...
\include {chapn}
\numberofappendices = 3
\appendix
\include {appa}
\include {appb}
\include {appc}
\bibliographystyle {siam}
\bibliography {\jobname}
\end {document}
```

The `uthesis2e` class file, although modeled on the \LaTeX 2.09 `report` style, is eclectic, and uniquely intended for use at just one institution.

Also, it has far too little internal documentation, making it hard to understand, maintain, and modify when the thesis layout regulations are occasionally changed.

We now view that file as **permanently frozen**, and all future changes to its formatting are to be done with separate packages included *later* in the \LaTeX input stream.

As long as students did not need nondefault formatting, the original `uuthesis2e` class file worked reasonably well. However, modern documents usually need much more, and by the \TeX Live 2015 release, the number of available \LaTeX packages had grown to almost 4300, and the count of bibliography styles to about 400. Each of the \LaTeX packages usually has a main design function, such as to change the default font families, or to typeset figures defined in various graphics-file formats, and to do so, the package has to do two things: (1) define its own internal macros with unique names that do not collide with names defined by users or by other packages, and (2) redefine some internal macros used by the \LaTeX core.

Point (2) is the root of a huge problem: when multiple packages are included, the final typeset appearance is likely to depend on the *order* in which those packages are read by \TeX , because the most recent definition of a macro is used when it is finally executed. Because of the limited size of computer memories when \TeX was designed (it could run on systems with as little as 0.25MB of memory!), \TeX does not keep a record of where each macro definition comes from.

For the University of Utah Thesis Office requirements, point (2) takes a lot of work to control. Even though there may be correct macro and parameter definitions in the class file that \LaTeX reads on startup, many of those definitions have been changed by the time typesetting starts, sometime after the `\begin {document}` command is processed.

As a result, considerable care is needed in preparation of the thesis input file and its package order. It is *unavoidable* that there must be repairs made after all other packages are included, and those repairs are supplied by a new package file whose name is `uuthesis-2016-d.sty`, but both that year and the version letter are certain to change in the future.

In addition, during the development of that fixup file, the maintainers felt that it is essential to provide some additional facilities in a set of optional style files that must be input *after* the fixup file, because they may need to make further redefinitions of internal macros to implement particular features.

3 The 2016 thesis file structure

THE COMMON FILE — A PLAGUE! TRIBUNES FOR THEM!

WILLIAM SHAKESPEARE, *Coriolanus* (1605).

The Internet sites given earlier provide access to individual files, and bundles thereof, and the sample thesis file is heavily commented to aid in its understanding. However, those comments also obscure its basic structure, so here is what \LaTeX actually processes from a test thesis file, which we can easily see by stripping comments and empty lines:

```
% egrep -v '%|^ *$' thesis.tex
\documentclass [11pt,Chicago] {uuthesis2e}
\let \proof = \relax
\let \endproof = \relax
\usepackage {amsthm}
\usepackage {mathpazo}
\fourlevels
\usepackage {amsmath}
\usepackage {amssymb}
\usepackage {bm}
\usepackage {citesort}
\usepackage {graphicx}
\usepackage {longtable}
\usepackage {multirow}
\usepackage {tikz}
\usetikzlibrary {decorations.markings}
\usepackage {uuthesis-2016-d} % MANDATORY package
\usepackage {uuthesis-pageheaders}
\usepackage {uuthesis-color-headings}
\usepackage {coloralias}
\usepackage {mythesis}
\author {Jane Doe}
\title {INTEGRAL ESTIMATION IN QUANTUM PHYSICS}
\thesistype {dissertation}
\dedication {For my parents, Alice and Bob.}
\degree {Doctor of Philosophy \\
in \\
Mathematical Physics}
\approvaldepartment {Mathematics}
\department {Department of Mathematics}
\graduatedean {Alice B. Toklas}
\departmentchair {Petrus Marcus Aurelius Featherstone-Hough}
\committeechair {Cornelius L{'a}nczos}
\firstreader {Hans Bethe}
\secondreader {Niels Bohr}
\thirdreader {Max Born}
\fourthreader {Paul A. M. Dirac}
\chairtitle {Professor}
\submitdate {May 2016}
\copyrightyear {2016}
```

```

\chairdateapproved      {17 Feb 2016}
\firstdateapproved      {17 Feb 2016}
\seconddateapproved     {17 Feb 2016}
\thirddateapproved      {17 Feb 2016}
\fourthdateapproved     {17 Feb 2016}
\begin{document}
\frontmatterformat
\titlepage
\copyrightpage
\dissertationapproval
\setcounter {page}      {2} % UofU demands abstract on p. iii: start one lower
\preface {abstract} {Abstract}
\dedicationpage
\tableofcontents
\listoffigures
\listoftables
\optionalfront {Notation and Symbols} {\input {notation}}
\maintext           % Start normal page numbering: parts and chapters follow.
\pagestyle {headings} % NEW for 2016
\include {chap1}
\include {chap2}
\include {chap3}
\include {chap4}
\numberofappendices = 3
\appendix
\include {appa}
\include {appb}
\include {appc}
\bibliographystyle {siam}
\bibliography {\jobname}
\end {document}

```

4 Colored section headings

MY HANDS ARE OF YOUR COLOUR;
BUT I SHAME TO WEAR A HEART SO WHITE.

LADY MACBETH IN WILLIAM SHAKESPEARE'S *Macbeth* (1599).

Default L^AT_EX classes normally set all text in black, but color is an important feature of the human visual system, and restricting documents to black-and-white appearance is as undesirable today as black-and-white printers and black-and-white television. Color, when carefully used, can make documents more attractive, and importantly, enhance document readability by providing subtle visual clues about the purpose of the text.

Although it was not considered in the original design of \TeX , color is quite easy to incorporate in \LaTeX documents. The standard `color` and `xcolor` packages provide the necessary support, and the local style files `coloralias.sty` and `rgb.sty` make it easy to work with named colors.

The basic `\color{name}` command works like the original \LaTeX font-changing commands — `\bf`, `\em`, `\it`, `\rm`, `\sc`, `\sf`, `\sl`, and `\tt` — once issued, the change remains in effect until end of document, or end of group, whichever occurs first. In \LaTeX , groups are delimited by braces (*excluding* the output braces in macro definitions), or by `\begin{env}` ... `\end{env}` environment wrapper commands. The page background remains unchanged, but text is set in the specified color in the current font.

By contrast, the `\textcolor{name}{text}` command limits the scope of the color change to just the second argument.

In both commands, colors are specified by name, and we show shortly how to define them.

Fonts themselves know nothing of color; color is applied only at the page-rendering stage, so the mechanism works for all fonts.

A new optional style file is available for inclusion after the fixup package, `uthesis-YYYY-L`, like this:

```
\usepackage {uthesis-color-headings}
```

That package modifies internal macros to typeset all sectional headings in a different color, the default being the official University of Utah red color. It also typesets figure and table labels in the heading color. In addition, it typesets the labels of all theorem-like environments defined by the \LaTeX `\newtheorem` command in a theorem color whose default is a blue of the same intensity as Utah red.

Both of those colors can be changed after package inclusion with definitions like this:

```
\definecolor{utahheadingcolor} {rgb} {0.700, 0.000, 0.000} % darker Utah red
\definecolor{utahtheoremcolor} {rgb} {0.490, 0.149, 0.804} % purple4
```

Those two define lists of red, green, and blue intensities, each on a scale of zero to one.

More conveniently, it is also possible to select colors by name from the palette of more than 650 defined in the X11 window system on Unix computers, and displayable with the `showrgb` command. Those color definitions have been captured and converted to \LaTeX commands in the `rgb.sty` style file that is automatically included by higher-level thesis styles.

You can then write your redefinitions like this:

```
\definecoloralias{utahheadingcolor}{orangered3}
\definecoloralias{utahtheoremcolor}{steelblue4}
```

Many Unix systems have tools that allow you to mix colors dynamically on the screen: `kcolorchooser` and `kcoloredit` are two common examples. If you manage your own Unix desktop or portable computer, you can probably search the list of available packages for those containing the text *color* to find other such tools.

There is extensive documentation on the use of color, both for text and for tabular cell and page backgrounds, in the \LaTeX companion volumes:

- *The \LaTeX Companion* (second edition, Addison-Wesley, 2004, ISBN 0-201-36299-6);
- *The \LaTeX Graphics Companion* (second edition, Addison-Wesley, 2008, ISBN 0-321-50892-0); and
- *The \LaTeX Web Companion* (Addison-Wesley, 1999, ISBN 0-201-43311-7).

5 Running page headers

WRITING A BOOK IS LIKE DOING A MARATHON:
BOTH REQUIRE YEARS OF PREPARATION,
BOTH INVOLVE A LOT OF RUNNING,
AND BOTH END IN COMPLETE EXHAUSTION!
BUT AFTER SOME REST, WE ARE READY AGAIN.

The original `uthesis2e` class file eliminated support for running page headers that carry chapter and section titles on alternate pages, and instead adopted a plain style that sets only a bare page number in the upper-right corner of every page, based on the common practice of single-sided printing of thesis documents.

However, that plainness also impedes quick navigation through the document. The standard modern \LaTeX styles for reports and books normally provide for incorporation of the current chapter or section title in the page header, and you can now get that effect with a command like this:

```
\usepackage{uthesis-pageheaders}
```

The low-level \LaTeX output routine adds header and footer lines, and pending footnotes, to the page image shipped to the output file. It checks whether two-sided printing is in effect, and if so, uses the parity of the current page number to determine whether the page is even (on the left of a two-page spread) or odd (on the right), and uses the contents of even and odd header and footer macros accordingly. Otherwise, it uses only the contents of the *even* macros.

The above style file therefore has to redeclare printing to be two-sided, even though it is not, in order to be able to place the chapter heading on odd pages, and the section heading on even pages.

Often, even and odd pages are set with different margins, with the inner margin of a two-page spread wider than the outer margin, to allow additional space for book binding. For University of Utah theses, however, those margins are identical, so no harm arises from declaring that two-sided printing is in effect.

Because page headings have to fit on one line, if you have some long sectional headings, you need to supply a square-bracketed argument with a shorter form that is used for the page headers and for the table of contents:

```
\section [Dirac-like antiparticles in fields]
        {Dirac-like antiparticles in the presence of electric
        and magnetic fields}
```

The same advice applies to the `\caption` macro: its bracketed short form for the lists of figures and tables should not be longer than about 2/3 of a line, whereas its braced long form could run for several paragraphs if a lot of description is needed to clarify a figure or table.

6 Other useful packages

AND TO WHAT PURPOSE WILL ALL THIS BUSTLE OF LIFE,
THESE AGITATIONS AND EMOTIONS OF THE HEART HAVE CONDUCTED,
IF IT LEAVE BEHIND IT NOTHING OF UTILITY,
IF IT LEAVE NO TRACES OF IMPROVEMENT?

MARK TWAIN, *The \$30,000 Bequest* (1906).

In the interest of improving document navigation and readability, there are a few other packages that are worth using.

If you use a numeric citation style, with inline citation lists generated by a command like `\cite{lab1,lab3,lab17}`, it is desirable to have the list of citation numbers sorted numerically, with ranges replacing two or more consecutive page numbers. That facility is provided by this command in your document preamble:

```
\usepackage {citesort}
```

You can help your readers by including cross references between different parts of your document:

In the figure with green lizards, ... % BAD: where is that figure?

In Figure 3, ... % STILL BAD: 3 is a magic number
% that is likely to be wrong the

```

% next time you update the document!

In Figure \ref{green-lizards}, ... % GOOD: but may be inconsistent

\newcommand {\figlabel} [1] {\label{fig:#1}}
\newcommand {\figref} [1] {Figure~\ref{fig:#1}}
In \figref{green-lizards}, ... % BETTER: uniform notation and
% label namespace protection

```

However, the reader is still faced with a reference to a figure that may be many pages away in the document, and thus, hard to find.

A good solution is provided by a standard \LaTeX package that is documented in *The \LaTeX Companion*:

```
\usepackage {varioref}
```

With that package, you can change your cross-referencing support like this:

```

\newcommand {\figlabel} [1] {\label{fig:#1}}
\newcommand {\figref} [1] {Figure~\ref{fig:#1} \vpageref{fig:#1}}
\newcommand {\shortfigref} [1] {Figure~\ref{fig:#1}}

In \figref{green-lizards}, ... % BETTER: uniform notation and
% label namespace protection

```

Now, the typeset output contains phrases like *Figure 3.1 on page 23* or *Figure 3.1 on the next page*. If the referenced object is found on the same page, then `\vpageref` produces no output, so you get simply *Figure 3.1*. In every case, the reader then knows exactly where to find the referenced matter.

The suggested macros intentionally prefix your labels with an object type, so that you don't accidentally mix types, by typesetting *Table 5.3* when you really meant *Figure 5.3*.

You might even need a fancier interface if you reference ranges of objects:

```

\newcommand {\figrefrange} [2] {Figure~\ref{fig:#1} \vpageref{fig:#1}
through
Figure~\ref{fig:#2} \vpageref{fig:#2}}

```

You should define similar private reference macros for each kind of object that you need to cross reference: for example, `\chapref`, `\sectref`, `\eqref`, and `\tabref`, for chapters, sections, equations, and tables.

If you insist on using explicit `\ref` commands in your input prose, then you have to proofread your output carefully to avoid inconsistencies like *Fig. 3.1*, *fig. 3.1*, *Figure 3.1*, and *figure 3.1*. Such accidents are far too common in many existing \LaTeX documents.

The reason for providing a page-number-free variant, `\shortfigref`, is for those rare occasions where including a page number would produce an overfull line, or an unwanted page break, and you are willing to sacrifice a bit of your readers' convenience.

7 Final formatting tweaks

THEN WE SAT DOWN TO POLISH OFF THE PERSPIRATION AND
ARRANGE ABOUT WHAT WE WOULD DO WITH HIM WHEN WE GOT HIM.
HARRIS WAS FOR CONTRIBUTING HIM TO THE BRITISH MUSEUM;
BUT I WAS FOR MAILING HIM TO HIS WIDOW.
MARK TWAIN, *A Tramp Abroad* (1880).

In cases such as those described in the previous section, you may still be able to include a page reference, or avoid an overfull box, by reworking your prose to make it shorter. You can also help T_EX to hyphenate words that its powerful language-dependent rules do not handle properly:

```
% in document preamble:
\hyphenation{
  Bern-oulli
  hexa-dec-i-mal
  lower-case
  Mar-cin-kie-wicz
  Na-ray-a-nan
  Papa-christ-o-pou-los
  Poch-hammer
  Post-Script
  trap-e-zoid
  Ur-i-be
  upper-case
  Ver-schae-ren
  white-space
}

% discretionary hyphen in prose for possible one-time use
The compiler ignores comments and white\ -space.
```

T_EX's hyphenation algorithm was worth a Stanford University doctoral degree, and it is so good that few users ever need to assist it by such techniques as we showed.

However, technical documents are often full of peculiar words, compound trade phrases, and words from foreign languages, so thesis authors are likely to need at least a few instances of hyphenation help.

However, do not bother doing so until your writing is completely finished, and you are engaged in final adjustments to improve bad line and page breaks.

Although \TeX warns about overfull boxes in the output log file, they are sometimes hard to spot in screen document viewers. It is easy to make them stand out during draft production with a command like this in your document preamble:

```
\overfullrule = 100pt
```

Every overfull horizontal box is then flagged with a wide black rule in the right margin. You can comment out that line, or reset its value to `0pt`, for your final output.

If you want to ignore *slightly* overfull boxes, you can do so like this:

```
\hfuzz = 3pt
```

\TeX then does not warn about lines that are less than `3pt` overfull, nor does it mark them with a marginal rule.

\TeX is normally rather concerned about overfull boxes, and both \TeX and \LaTeX have default values of the horizontal and vertical fuzz parameters set to `0.1pt`, which is less than most humans can spot on a printed page.

If you have a paragraph that seems not to be amenable to either additional hyphenation help, or rewriting, and still produces an overfull box, you can ask \TeX to be less fussy in its typesetting:

```
\begin {sloppypar}
... problem prose here ...
\end {sloppypar}
```

```
\begin {rm}
\emergencystretch = 3pt
... problem prose here ...
\end {rm}
```

```
\begin {rm}
\tolerance = 500
... problem prose here ...
\end {rm}
```

```
\begin {rm}
\looseness = +1 % try to make paragraph one line longer
... problem prose here ...
\end {rm}
```

```

\begin {rm}
  \looseness = -1  % try to make paragraph one line shorter
  ... problem prose here ...
\end   {rm}

```

The `rm` environment, which selects a roman font, simply serves as a convenient wrapper that limits the scope of the parameter changes. If you want to be more general and not risk a font change, you should instead define your own wrapper environment like this:

```

\newenvironment {dummy} {\begingroup} {\endgroup}
...
\begin {dummy}
  \looseness = +1  % try to make paragraph one line longer
  ... problem prose here ...
\end   {dummy}

```

One thing that you should *almost never do* in a thesis document is to insert explicit horizontal and vertical spacing with \LaTeX `\hspace{}`, `\hspace*{}`, `\vspace{}`, or `\vspace*{}` commands, or low-level \TeX `\kern`, `\hskip`, or `\vskip` commands. Nor should you *ever* use consecutive ties (`~ ~ ~ . . .`) — \TeX 's unbreakable spaces — to achieve horizontal spacing. If you think that you need such adjustments, you are probably wrong, and your document formatting will not be robust against further changes in your input files. The whole point of \LaTeX markup is that the formatting commands have the *entire responsibility* of getting spacing right, and authors should just concentrate on getting their prose right.

8 Creating a subject index

IN THE BAD OLD DAYS, THE INDEX WAS A LIST OF
PROHIBITED BOOKS; MAY WE NOW, IN A MORE
ENLIGHTENED AGE, BAN BOOKS WITHOUT INDEXES?

— STEPHEN JAY GOULD, *An Urchin in the Storm* (1987).

A thesis or dissertation is essentially a book, and nonfiction books deserve indexes to aid readers. Fiction books are normally read sequentially, and have content that is often ephemeral. By contrast, nonfiction books, and scientific articles, are often read in unpredictable order, and at different times, and their content ought to be longer lived. A reader might start with the table of contents, then look in the index, then jump to chapter 3 and read its introduction and conclusion before deciding whether to read the entire chapter. Some time later, the reader might return to the document to find a previously-read topic, and then, the index is probably the most valuable navigation aid, along with colored sticky tabs left in the document on a previous reading.

Because a thesis is often a student's first large writing effort, and more such projects are likely to follow in future, writing a thesis is a good opportunity to learn how to create a useful index. This section supplies most of what you need to know to do so.

Standard \TeX distributions are always accompanied by a separate program, `makeindex`, that assists in the preparation of a document index. Its job is to read a random list of index entries, and assemble them into a lexicographically-sorted and merged list of index topics, each with a numerically sorted list of page numbers where that topic is discussed. Consecutive page numbers in each list are replaced by ranges. Additional markup is supplied to allow \TeX to typeset the index.

It is also possible through additional index-entry markup to distinguish certain page numbers as being more important than others, usually by setting them in a **bold font**.

Briefly, a \TeX command `\index{gnu}` writes out to the associated `.idx` file a command `\indexentry{gnu}{23}` that records the indexed phrase and its page number. Running `makeindex` on that file produces a \TeX input file with a name ending in `.ind` that contains the sorted and merged index entries and letter headings.

The \TeX user then prepares a top-level file that looks like this:

```
\documentclass {somename}
\usepackage{makeidx}
\makeindex           % comment out to suppress index generation
\begin {document}
...
\printindex         % index is normally last typeset material
\end   {document}
```

Here is a fragment of the \TeX index file produced by `makeindex` for a recent large book on mathematical software:

```
\begin{theindex}
  \newindexgroup{Symbols}
  ...
  \indexspace
  \newindexgroup{A}
  ...
  \item Abel, Niels Henrik (1802--1829)\quad \pagebf{10}, 825
  ...
  \item ACM Turing Award\quad 1007, 1279, 1285
  \item algorithm
    \subitem argument reduction (exact)
      \subsubitem implementation\quad 344
      \subsubitem retrospective\quad 361
      \subsubitem testing\quad 361
      \subsubitem theory\quad 340
```

```

...
\indexspace
\newindexgroup{B}
...
\item Babbage, Charles (1791--1871)\quad \pagebf{1205}
\item Babylonians\quad 11, 405
...
\indexspace
\newindexgroup{Z}
...
\item zeta function\quad 76, 410, \pagebf{775}
  \subitem Hurwitz generalization of Riemann\quad 781
...
\item zoetrope\quad 33
\item \textrm {Zuse} (vendor)\quad 1262, 1266, 1295
\item Zuse, Konrad (1910--1995)\quad \pagebf{1266}
\item Zeta function (Jacobian)\quad 904
\end{theindex}

```

The default definition of the `theindex` environment typesets the index in two columns. If you have a lot of short index entries, then you *might* be able to typeset it in three or even four columns with the help of the standard \LaTeX `multicol` package and its `multicols` environment as described in *The \LaTeX Companion*. This document's author has done just that for at least three large books; feel free to contact him for further details.

If your index entry phrases are long, you can exploit the fact that the index is just a table that readers skim quickly to find a topic of interest, and set the index in a smaller font:

```

\begin{small}
  \printindex
\end{small}

```

For University of Utah theses, the top-level procedure for generating and typesetting an index is similar to the example above:

```

\documentclass [11pt,Chicago] {uthesis2e}
...
\usepackage {uthesis-2016-d} % MANDATORY package
\usepackage {uthesis-index}
...
\begin {document}
...
\printindex % index is normally last typeset material
\end {document}

```

For convenience, the `\makeindex` command is issued by the style file; computers are now fast enough that the savings of suppressing index-file generation are not measurable.

The basic job steps to create the index then typically look like this, assuming PDF file generation:

```
pdflatex thesis.tex
bibtex thesis
makeindex -c -s thesis.ist thesis.idx
```

That triple of commands needs to be repeated two or more times to reach consistency of cross referencing, table of contents, lists of figures and tables, bibliography, float positioning, and index.

The `-c` option on the last command says that leading and trailing whitespace are to be ignored, and runs of whitespace are to be compressed to a single space. That way, similar, but not identical, index entries

```
\index_{gnu_hair}
\index_{gnu_hair}
\index_{gnu_hair}
\index_{gnu_hair}
```

would be collapsed to a single index phrase.

The `-i` option specifies an index style file. We use a simple one with these short contents:

```
heading_prefix      "  \newindexgroup{"
heading_suffix      "}\nopagebreak"
headings_flag       1
delim_0             "\quad "
delim_1             "\quad "
delim_2             "\quad "
```

The first two lines provide a wrapper macro for the index section letter, and that macro is defined in the `uthesis-index` package. The remaining options are completely documented in the Unix manual pages for the program: run the command `man makeindex` to read them.

To prepare an index, it is usually worthwhile to apply a highlighter pen to a nearly-final printed copy of the document, marking phrases in the text that are likely to be useful for the index. You can then go through your \LaTeX files and insert suitable indexing commands. Because it is common to have identical phrases in the document prose and its index, it is helpful to define a macro that generates both:

```
\newcommand {\X} [1] {#1\index {#1}}
```

Notice that definition: it is *essential* that no space occurs before the `\index` command, because it would then be possible for a page break

to occur between the phrase and the index entry, and the indexed page number would be off by one. The author of this document has created `emacs` text-editor support that allows adding the `\X{...}` wrapper with a simple sweep of the mouse.

For inline indexing, it is helpful to keep all `\index` commands on separate lines, because they can then be easily found by search commands, like the Unix `grep` utility:

```
\section {The \bioname{E. coli} bacterium}%
\index {E. coli bacterium@\bioname{E. coli} bacterium}
```

The final percent symbol starts a \TeX comment that runs through the end of line, *including* the newline character, and all leading whitespace on the *next* line: thus, space between the phrase and the indexing command is eliminated, yet the command remains easily visible.

That example shows another feature of `makeindex` that you almost certainly need in technical documents: separation of the sorted indexing phrase from its typeset appearance. In general, you can write

```
\index {sort-key@typeset phrase}           % alternate sorting
\index {main phrase!sub phrase}           % 2-level entry
\index {main phrase!sub phrase!subsub phrase} % 3-level entry
```

Latin names used in biology are normally set in emphasized text, but we used a private macro `\bioname{...}` instead of `\emph{...}` for two reasons: markup should always indicate what things *are*, not how they *look*, and having a special macro name gives us additional flexibility. You might therefore have in your private style file commands like one of these:

```
\newcommand {\bioname} [1] {\emph{#1}\index{#1@\emph{#1}}}
\newcommand {\bioname} [1] {\emph{\color{blue}#1}\index{#1@\emph{#1}}}
```

That would emphasize the argument text, and also generate an index entry for it. Provided that your markup is consistent, your index entries are then both consistent and complete.

In multilevel indexes, you can have separate sort keys for each level, like this:

```
\index {sort-key-1@phrase-1}
\index {sort-key-1@phrase-1!sort-key-2@phrase-2}
\index {sort-key-1@phrase-1!sort-key-2@phrase-2!sort-key-2@phrase-2}
```

In each case, the prefix `sort-key-n@` can be dropped if the phrase itself would sort properly.

Finally, you often want to have cross references in the index, so that synonyms for a phrase redirect the reader to a common index entry with a page-number list:

```

\index {DCT|see{Discrete DCT Transform}}
\index {FFT|see{Fast Fourier Transform}}
\index {coli, E.@\emph{coli, E.}|see{E. coli}}
\index {Escherichia coli@\emph{Escherichia coli}|see{E. coli}}
\index {transform|seealso{Discrete DCT Transform}}
\index {transform|seealso{Fast Fourier Transform}}

```

Such entries do not generate a page number, so it *does not matter* where you put them in your document. Two good places are the private style file where you record your own macro definitions, and just before the `\printindex` command. Either way, those entries are then kept together so that you can easily find them and ensure their consistency and completeness.

Here are some examples of simple multilevel indexing (such commands would, of course, be sprinkled at appropriate places in your \LaTeX document files):

```

\index {GNU Project}
\index {gnu}

\index {gnu!diet}
\index {gnu!diet!wet season}
\index {gnu!diet!dry season}

\index {gnu!hair}
\index {gnu!hair!color}
\index {gnu!hair!DNA analysis}
\index {gnu!hair!texture}
\index {gnu!hair!thickness}

\index {gnu!predators}
\index {gnu!predators!crocodiles}
\index {gnu!predators!hyenas}
\index {gnu!predators!lions}

\index {African ungulate|see{gnu}}
\index {ungulate|see{gnu}}
\index {ungulate|see{impala}}
\index {ungulate|see{kudu}}
\index {ungulate|see{springbok}}
\index {wildebeest|see{gnu}}

\index {free software|seealso{GNU Project}}

```

The default heading supplied for the index by the `uthesis-index` package is **SUBJECT INDEX**. You can easily change that name anywhere after the package inclusion with a command like this:

```
\renewcommand {\indexname} {TOPIC INDEX}
```

The expansion should be an *UPPERCASE* phrase to match the chapter heading style in University of Utah theses.

If you wish to distinguish major and minor index entries, you can do so with additional commands defined in your private macro style file like this:

```
\newcommand {\mainindex} [1] {\index{#1|mainpage}}
\newcommand {\mainpage} [1] {\textbf{#1}}
```

You then replace `\index` by `\mainindex` in those places that you wish to emphasize in the index page-number lists.

In that example, notice carefully how the `mainpage` string is used: it appears in the index argument prefixed by a vertical bar, and followed by nothing. However, after `makeindex` runs, that name gets converted to a \TeX macro in the `.ind` file that wraps a page number like this:

```
\item ... \quad 187, 213, \mainpage{321}, 375, 513
```

Because you control the appearance of those special numbers by your definition of the `\mainpage` macro, you could instead highlight them in color:

```
\newcommand {\mainpage} [1] {\textbf{\color{utahred}#1}}
```

If you have lots of index entries, it might even be useful to create two indexes: a short one with just the main page numbers, and the second the complex index. You could easily do that by extracting the main entries into a second file, and creating an index for it:

```
grep mainpage thesis.idx | makeindex -c -i thesis.ist > main-thesis.ind
```

Your top-level \TeX file might then end like this:

```
\renewcommand {\indexname} {MAIN TOPIC INDEX}
\input {main-\jobname.ind}
\renewcommand {\indexname} {FULL TOPIC INDEX}
\printindex
\end {document}
```

Separation of index entries into groups, each with a shorter index, can be helpful to readers. You might create separate indexes of biological names, book and article titles, computer programs, named theorems, personal names, and so on. Their entries would normally also be recorded in the full index, but you could also remove them from that index by simple filtering commands that leverage distinctive index-entry markup like this:

```

### create short alias for longer command
MAKEINDEX="makeindex -c -i thesis.ist"

grep bioname      thesis.idx | $MAKEINDEX > bioname.ind

grep booktitle    thesis.idx | $MAKEINDEX > booktitle.ind

grep personalname thesis.idx | $MAKEINDEX > personalname.ind

grep programname  thesis.idx | $MAKEINDEX > programname.ind

### remove the four special index entries from the master index
egrep -v 'bioname|booktitle|personalname|programname' thesis.idx |
  $MAKEINDEX > thesis.ind

```

Your top-level \LaTeX file then needs only two more lines for each of your specialized indexes, as illustrated earlier for main and full topic indexes.

Some final points on indexing are worth noting:

- Proofread your index phrases carefully to avoid instances of similar, but not identical, entries differing in singular and plural forms, or in nonessential lettercase. Unix commands like these can produce a temporary file of sorted index commands that you can quickly review for such mistakes:

```
fgrep '\index' *.ltx *.sty *.tex | sort -f > foo
```

- When you are creating index entries, think of your readers, and ask yourself *What phrases would they be looking for in my index?*, and *What are common synonyms for the phrases that I am indexing?*
- For heavily-used technical books, it is not unreasonable for the index to be 5% to 10% of the length of the book. For example, the *Chicago Manual of Style* (16th edition, University of Chicago Press 2010, ISBN 0-226-10420-6) is *xvi* + 1026 pages long, and its index spans 102 pages. It also has an entire chapter devoted to document indexing.
- There are also a few books devoted entirely to the topic of indexing, including Kurt Ament's *Indexing: a nuts-and-bolts guide for technical writers* (William Andrew Publishing, 2001, ISBN 0-8155-1481-6) and Ruth Cross' *Indexing books* (Word Guild, 1980, ISBN 0-9602524-1-X).

9 What is unlikely to work

ADIEU, TILL THEN; THEN, FAIL NOT.

WILLIAM SHAKESPEARE, *All's well that ends well* (1604).

One package that should be strictly avoided is the [geometry](#) package, because it is intended for redefining page layout, and that layout is already fixed by university regulations.

An important and popular package that is used for generating hypertext links in PDF files produced by [pdflatex](#) is [hyperref](#). Its documentation strongly recommends that it be included last, because it redefines a large number of \LaTeX internal macros, including those for section headings, cross references, bibliography item references, equations, figures, tables, and Internet URLs.

Unfortunately, those redefinitions almost certainly conflict with the many redefinitions made by the University of Utah class and style files, and are likely to lead to rejection of your thesis on the grounds of improper formatting.

Support for hypertext references in future theses is highly desirable, but not easy to accomplish. \TeX itself is about 20,000 lines of code. The [uthesis2e](#) class file is about 2100 lines, and the 2016 style-file extensions add another 900 lines. By comparison, the [hyperref](#) package alone consists of more than 28,000 lines of code. The entire \LaTeX corpus of class and style files is more than 2,000,000 lines of code, and the bibliography style files contribute another 600,000 lines of code. In addition, the \TeX macro language is one of the most difficult computer languages to program, and there are many hidden connections between formatting parameters that can make it excruciatingly hard to predict behavior, and debug erroneous formatting actions.

It appears that the only reliable way for future University of Utah theses to have hypertext references that do not affect formatting will be for a professional designer to take on the large job of writing an entirely new class file that is intimately linked to a modified [hyperref](#) package. That work could be substantially reduced, however, if the university thesis formatting regulations were made more flexible, and less prescriptive about vertical and horizontal spacing. Such ideas have been proposed, but adoption of significant changes in a large organization takes much time and much discussion, so it is impossible to predict when, or if, such a new redesign might be undertaken and completed.