The **ETbb** package—Edward Tufte’s version of Bembo

Michael Sharpe

**Background**

The fonts in this package were derived ultimately from the collection of fonts commissioned by Edward Tufte for his own books, and released in 2015 as ET-Bembo under the MIT license. (The sources for that collection were fonts using the family name ET-book.) That collection was enhanced in 2019 under the name \texttt{XETBook} by Daniel Benjamin Miller, and it is his package which was the starting point for \texttt{ETbb}, where the bb denotes the Berry abbreviation for Bembo. The final section of this document makes a detailed comparison with the earlier \texttt{fbb} package, which is also Bembo-like, derived from \texttt{CarDo}. The most significant differences are that \texttt{ETbb} has a regular upright that is about 20\% darker than the corresponding \texttt{fbb}, and its ascender height is noticeably less. These differences make \texttt{ETbb} have a less spindly appearance that is closer in spirit to the print produced by traditional metal versions of Bembo.

**Package properties**

The package makes a number of changes to the \texttt{XETBook} fonts:

- The released version of ET-Bembo lacks kerning tables—a serious omission—rectified in \texttt{ETbb}.
- The scale has been increased by 3.36\% so that the x-height of the upright regular face is 431, very close to Computer Modern and Libertine.
- The lining figures in some faces were reduced so as to be a bit less than the cap-heights.
- The lining figures in \texttt{XETBook} were proportional rather than tabular. I’ve added new tabular lining and old-style figures.
- Added superior letters and figures to all faces. E.g., `\texttt{su(ABCabc123) renders as ABCabc123}`.
- Added inferior figures to all faces with baseline at \texttt{-112em}.
- Added denominator figures to all faces with baseline at \texttt{0em}.
- The originals comprised glyphs in the Adobe Standard Encoding, forming a rather sparse subset of the T1 encoding. I’ve added accented and composite glyphs that provide complete coverage of the T1 encoding as well as many glyphs required in the orthography of a number Eastern European countries.
- Prior to version 1.02, coverage of TS1 encoding was meager. The coverage is now close to full. (See the table at the end of this document.)
- Small caps have been added to all faces.
• There is a new glyph for the German capital sharp S (ß, großes eszett, U+1E9E), approved in 2017 for optional use in German orthography. Small cap versions are also provided.

• The glyph capital P has been changed from its default closed shape, as used in almost all modern digital renderings of Bembo, to the more historically accurate open shape. See, for example, the reproduction of Pietro Bembo’s De Aetna at
  (A higher resolution rendering of a two-page sample is available from https://upload.wikimedia.org/wikipedia/commons/8/89/De_Aetna_1495.jpg.)

Package options and macros

This package has most of the same features and options as the fbb package and even includes the altP option, though that has no effect because the alternate P shapes in fbb are the default in ETbb.

In the original XETBook, the dollar and cent currency symbols were oldstyle. I’ve added the new style symbols and made them the default, but option osdollar to ETbb changes back to the oldstyle symbols.

Text figures may be selected from four types:

• Proportional lining (LF), selected by options lining, proportional [or p]; (lining, or lf, is the default figure style;)
• Tabular lining (TLF), selected by options lining [or lf], tabular [or t]; (tabular is the default figure alignment;)
• Proportional oldstyle (OsF), selected by option oldstyle [or osf], proportional [or p];
• Tabular oldstyle (ToSF), selected by options oldstyle [or osf], tabular [or t].

The package also defines macros that allow you use alternate figure styles locally:

<table>
<thead>
<tr>
<th>Macro</th>
<th>result</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textlf{0123456789}</td>
<td>0123456789</td>
<td>print 0123456789 in proportional lining figures</td>
</tr>
<tr>
<td>\texttlf{0123456789}</td>
<td>0123456789</td>
<td>print 0123456789 in tabular lining figures</td>
</tr>
<tr>
<td>\textosf{0123456789}</td>
<td>0123456789</td>
<td>print 0123456789 in proportional oldstyle figures</td>
</tr>
<tr>
<td>\texttosf{0123456789}</td>
<td>0123456789</td>
<td>print 0123456789 in tabular oldstyle figures</td>
</tr>
<tr>
<td>\textsu{0123456789}</td>
<td>0123456789</td>
<td>print 0123456789 in superior figures</td>
</tr>
<tr>
<td>\textin{0123456789}</td>
<td>0123456789</td>
<td>print 0123456789 in inferior figures</td>
</tr>
<tr>
<td>\textde{0123456789}</td>
<td>0123456789</td>
<td>print 0123456789 in denominator figures</td>
</tr>
</tbody>
</table>

The macro \textlf{123} is identical in effect to {\lfstyle 123}, and similarly for the other lining and oldstyle macros, while \textsu{123} has the same effect as {\sufi gures 123} and \textin{123} has the same effect as {\infigures 123}. If you prefer typing longer names, you may use \textinferior as a synonym for \textin, and similarly for \textsu. Likewise, \textde{123} has the same effect as {\defi gures 123} or {\denomfi gures 123}, and you may use \textdenom as a synonym for \textde. Note the difference in baseline between \textin and \textde: 123 versus 123.
The `\textfrac` macro constructs fractions using `\textsu` and `\textde` with baseline aligned with the text baseline. The behavior is quite configurable, there being three parameters available to control the kerns before and after the fraction solidus, and the amount to raise the resulting box. The three parameters are passed as options to ETbb, named

- `foresolidus % default value \-.05em`
- `aftsolidus % default value \-.08em`
- `raisefrac % default value \0em`

(The values should always be em units in order to behave correctly with respect to scaling.) If you were to change the default behavior with the option

`raisefrac=-112em`

you would get a fraction with the denominator’s baseline at the baseline of the `\textin` figures, namely \-.112em. All the above have to do with globally defined settings for `\textfrac`, but that macro allows one optional argument that can override the effect of `raisefrac`, with, e.g., `\textfrac[.1em][6]{11}` raising the fraction 6/11 by .1em instead of the default specified in the original options.

**Example:**
- `2\textfrac{17}{32}` renders as \(2\frac{17}{32}\) with default settings.

The `\textfrac` macro uses spacing control for each individual digit, one for the numerator and one for the denominator. It is optimized for regular weight, upright shape, but works satisfactorily in bold weight, upright shape. Italic shapes are not handled with any precision. Currently, the spacing settings are specified by two macros in ETbb.sty: `\tx@addNkern` and `\tx@addDkern` for numerator and denominator respectively. These may be redefined in your preamble after loading the ETbb package with a block like

```
\makeatletter
\renewcommand*{\tx@addNkern}[1]{% 
  \.
  \.
  \.
  \}
\renewcommand*{\tx@addDkern}[1]{% 
  \.
  \.
  \.
  \}
```

following the pattern of the definitions in `lt ETbb.sty`. You should be careful to follow those patterns precisely, as it is very easy to introduce inadvertent space characters in the output.

Option `supss` changes the form of footnote markers to use ETbb’s superior figures, unless you have redefined the meaning of `\textfootnote` prior to loading ETbb. For more control over size, spacing and position of footnote markers, use the `superiors` package: E.g.,

```
\usepackage[supstfm=ETbb-Regular-sup-t1]{superiors}
```

Option `sharpS` replaces $S$ in the T1 encoding by the new $U+1E9E$ glyph and replaces the small cap $ss$ by the small cap version of $U+1E9E$. Only figure-styles TLF, LF, OsF and T0sF are handled, and only in the
T1 encoding.

Option scosf forces the use of OSF figures in a small caps block, no matter what the default figure settings.

There is a scaled [or scale] option (e.g., scaled=.97) that allow you to adjust the text size against, say, a math package.

A suggested math companion

This text package works well with newtxmath with the libertine option, because the latter has italics of the same italic angle as ETbb and of very similar xheight and weight. If you have the MinionPro fonts (version 2.0 or higher) and have set them up with FontPro and the minion2newtx ctan package, then the minion option to newtxmath provides a very good math companion with better Greek letters than libertine. The suggested invocation for libertine math is:

\begin{verbatim}
% load babel package and options here
\usepackage[p,osf]{ETbb} % osf in text, tabular lining figures in math
\usepackage[scaled=.95,type1]{cabin} % sans serif in style of Gill Sans
\usepackage[varqu,varl]{zi4}% inconsolata typewriter
\usepackage[T1]{fontenc} % LY1 also works
\usepackage[libertine,vvarbb]{newtxmath}
%\usepackage[cal=boondoxo,bb=boondox,frak=boondox]{mathalfa}
\end{verbatim}

Here is a short sample based on this preamble:

The typeset math below follows the ISO recommendations that only variables be set in italic. Note the use of upright shapes for d, e and \pi. (The first two are entered as \mathrm{d} and \mathrm{e}, and in fonts derived from newtxmath or mtpro2, the latter is entered as \uppi.)

\textbf{Simplest form of the Central Limit Theorem:} Let \(X_1, X_2, \cdots \) be a sequence of iid random variables with mean 0 and variance 1 on a probability space \((\Omega, \mathcal{F}, \mathbb{P})\). Then

\[
\mathbb{P}
\left(
\frac{X_1 + \cdots + X_n}{\sqrt{n}} \leq y
\right) \to \Phi(y) := \int_{-\infty}^{y} \frac{e^{-t^2/2}}{\sqrt{2\pi}} \, dt \quad \text{as } n \to \infty,
\]

or, equivalently, letting \(S_n := \sum_1^n X_k\),

\[
\mathbb{E} f(S_n/\sqrt{n}) \to \int_{-\infty}^{\infty} f(t) \frac{e^{-t^2/2}}{\sqrt{2\pi}} \, dt \quad \text{as } n \to \infty, \text{ for every } f \in bC(\mathbb{R}).
\]

\textbf{Glyphs in TS1 encoding}

The layout of the TS1 encoded Text Companion font, which is rendered in regular style only, is as follows.
List of macros to access the TS1 symbols in text mode:
(The commented lines are in fbb but not ETbb.)

0 \capitalgrave
1 \capitalacute
2 \capitalcircumflex
3 \capitaltilde
4 \capitaldieresis
5 \capitalhungarumlaut
6 \capitalring
7 \capitalcaron
8 \capitalbreve
9 \capitalmacron
10 \capitalsdotaccent
11 \capitalcedilla
12 \capitalgonek
13 \textstraightbase
18 \textstraightdblbase
21 \texttwelveonedash
22 \textthreequartersemdash
23 \textcapitalcompwordmark
24 \textleftarrow
25 \textrightarrow
26 \t % tie accent, skewed right
27 \capitaltie % skewed right
28 \newtie % tie accent centered
29 \capitalnewtie % ditto
31 \textascendercompwordmark
32 \textblank
36 \textdollar
39 \textquotesingle
42 \textasteriskcentered
45 \textdblehyphen
47 \textfracsolidus
48 \textzeroldstyle
49 \textoneoldstyle
50 \texttwooldstyle
49 \textthreeoldstyle
50 \textfouroldstyle
51 \textfiveoldstyle
52 \textsixoldstyle
53 \textsevenoldstyle
54 \texteightoldstyle
55 \textnineoldstyle
60 \textangle
61 \textminus
62 \texttriangle
77 \textmho
There is a macro \textcircled that may be used to construct a circled version of a single letter using \textbigcirclearrowright. The letter is always constructed from the small cap version, so, in effect, you can only construct circled uppercase letters: \textcircled{M} and \textcircled{m} have the same effect, namely Ø.

**Usage with fontspec**

Because the package supplies a file named ETbb.fontspec whose contents list the otf files that correspond to each of Regular, Bold, Italic and BoldItalic, you may load ETbb with just

\usepackage{fontspec}
\setmainfont{ETbb}

Other than the usual choices of figure style, the only remaining choice available is through StylisticSet=2,
which substitutes the new Sharp S glyphs in place of the familiar $\beta$, SS and ss. See the table in the next section for details.

**Selection of the new Sharp S in LaTeX**

The following table summarizes the behavior of the text macros \texttt{SS}, \texttt{ss} and the macro \texttt{MakeUpperCase}.

<table>
<thead>
<tr>
<th>sharpS option</th>
<th>\texttt{ss}</th>
<th>\texttt{SS}</th>
<th>\texttt{MakeUpperCase}{\texttt{ss}}</th>
<th>\texttt{textsc}{\texttt{ss}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not set</td>
<td>$\beta$</td>
<td>SS</td>
<td>$SS$</td>
<td>ss</td>
</tr>
<tr>
<td>sharpS</td>
<td>$\beta$</td>
<td>$\beta$</td>
<td>$\beta$</td>
<td>$\beta$</td>
</tr>
</tbody>
</table>

In unicode TeX, the behavior laid out in the table above is achieved using \texttt{StylisticSet}=2.

**Detailed comparison with fbb**

The following picture, in which the units are approximately in bp, shows some of the differences between \texttt{ETbb-Regular} and \texttt{fbb-Regular}, the first scaled up by 10 and the second by 9.8 so that their x-heights (and Cap- heights) are the same. From the picture below you can note the following.

- The serifs are much more substantial in \texttt{ETbb}.
- The ascenders a considerably higher in \texttt{fbb}—in fact, by 50 em units. Those very tall ascenders make for poor positioning of quotes, superscripts and the like.
- Stems are a little thicker (by about 10%) in \texttt{ETbb}.
- There is a slight bowing out in the letter h and similar letters like n of \texttt{ETbb} that is not present in \texttt{fbb}, making for more visual interest, IMO. This would not be of any importance at small print sizes.
- Overall, \texttt{ETbb} has lower contrast (ratio of thickest to thinnest strokes) than \texttt{fbb}, making for a more uniform gray appearance on the printed page.

![Comparison of ETbb and fbb fonts](image)

The following page presents a comparison of a \texttt{ETbb} and \texttt{fbb} with identical text rendered in two columns. For me, there is no question that \texttt{ETbb} is the preferable font for document text.


