

Content-Type: text/enriched  
Text-Width: 70

<center><x-bg-color><param>blue</param><x-color><param>white</param><bold><fixed>enriched.el:</fixed></bold></x-color></x-bg-color>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>WYSIWYG rich text editing for GNU Emacs</bold></x-color></x-bg-color>

</center><bold><x-bg-color><param>blue</param><x-color><param>white</param>INTRODUCTION</x-color></x-bg-color></bold>

<indent>Emacs now has the ability to edit <italic>enriched text</italic>, which is text containing faces, colors, indentation, and other properties. This document is a quick introduction to some of the new features, and is also an example file in the <italic>text/enriched</italic> format.</indent>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>INSTALLATION and STARTUP</bold></x-color></x-bg-color>

<indent>Most of the time, you need not do anything to get these features to work. If you visit a file that has been written out in <italic>text/enriched</italic> format, it will automatically be decoded, Emacs will enter 'enriched-mode' while visiting it, and whenever you save it it will be saved in the same format it was read in.

If you wish to create a new file, however, you will need to turn on enriched-mode yourself:

<fixed><indent>M-x enriched-mode RET</indent></fixed>

Or, if you get a <italic>text/enriched</italic> file that Emacs does not automatically recognize and decode, you can tell Emacs to decode it (which also turns on enriched-mode automatically):

<fixed><indent>M-x format-decode-buffer RET text/enriched RET</indent></fixed></indent>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>WHAT IS ENCODED</bold></x-color></x-bg-color>

<indent>Here is the current list of text-properties that are saved; they are discussed in more detail below. Most of these can be added or changed with the "Text Properties" menu, available under the "Edit" item in the menu-bar, or on C-mouse-2 (Control + the middle mouse button).

<bold>Faces:</bold> <indent>default, <bold>bold</bold>, <italic>italic</italic>, <underline>underline</underline>, etc.</indent>

<bold>Colors:</bold> <x-color><param>red</param><x-bg-color><param>DarkSlateGray</param><indent>any</indent></x-bg-color></x-color><x-bg-color><param>DarkSlateGray</param><indent><x-color><param>orange</param>thing</x-color> <x-color><param>yellow</param>your</x-color><x-color><param>green</param> screen</x-color><x-color><param>blue</param> </x-color><x-color><param>light blue</param>can</x-color>><x-color><param>violet</param> display...</x-color></indent></x-bg-color>

<bold>Newlines:</bold> <indent>Which ones are real ("hard") newlines, and which can be changed to fit lines into the margins.</indent>

<bold>Margins:</bold> <indent>can be indented on the left or right.</indent>

<bold>Justification</bold> <indent>(whether lines should be flush with the left margin, the right margin, fully justified, centered, or left alone).</indent>

<bold>Excerpts:</bold><indent> <excerpt>"For quoted material."</excerpt></indent>

<bold>Read-only</bold> regions.

</indent>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>FACES and COLORS</bold></x-color></x-bg-color>

<indent>You can add faces either with the menu or with <fixed>M-g.</fixed> The face is applied to the current region. If you are using 'transient-mark-mode' and the region is not active, then the face applies to whatever you type next. Any face can have colors. If this is its lone attribute, the face is put on the color submenus of the "Text Properties" menu.</indent>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>NEWLINES and PARAGRAPHS</bold></x-color></x-bg-color>

<italic><indent>Text/enriched</indent></italic><indent> format distinguishes between <underline>hard</underline> and <underline>soft</underline> newlines. Hard newlines are used to separate paragraphs, or items in a list, or anywhere that must be a line break no matter what the margins are. Soft newlines are the ones inserted in order to fit text between the margins. The fill and auto-fill functions insert soft newlines as necessary, but hard newlines are only inserted by direct request, such as using the return key or the <fixed>C-o (open-line)</fixed> function.</indent>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>INDENTATION</bold></x-color></x-bg-color>

<indent>The fill functions also understand margins, which can be set for any region of a document. In addition to the menu items, which increase or decrease the margins, there are two commands for setting the margins absolutely: <fixed>C-c [ (set-left-margin)</fixed> and <fixed>C-c ] (set-right-margin)</fixed>.

You <indent>can change indentation at any point in a paragraph, which makes it possible to do interesting things like hanging-indents: this paragraph was indented by selecting the region from the second word to the end of the paragraph, and indenting only that part.</indent></indent>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>JUSTIFICATION</bold></x-color></x-bg-color>

<indent><nofill>Several styles of justification are possible, the simplest being <italic>unfilled.</italic>This means that your lines will be left as you write them. This paragraph is unfilled.</nofill>

<flushleft>The most common (for English) style is <italic>FlushLeft.</italic>This means lines are aligned at the left margin but left uneven at the right.</flushleft>

<flushright> <italic>FlushRight</italic> makes each line flush with the right margin instead. This paragraph is FlushRight.</flushright>

<flushboth><italic>FlushBoth</italic> regions, which are sometimes called "fully justified" are aligned evenly on both edges, so that the text on the page has a smooth appearance as in a book or newspaper article. Unfortunately this does not look as nice with a fixed-width font as it does in a proportionally-spaced printed document; the extra spaces that are needed on the screen can make it hard to read. </flushboth>

<center>

<bold>Center</bold>

Finally, there is <italic>center</italic> justification. The normal center-paragraph key, M-S, can be used to turn on center justification in enriched-mode.

M-j or the "Text Properties" menu also can be used to change justification.

Note that justification can only change at hard newlines, because that is the unit over which filling gets done.

**EXCERPTS**

This is an example of an excerpt. You can use them for quoted parts of other people's email messages and the like. It is just a face, which is the same as the 'italic' face by default.

**THE FILE FORMAT**

Enriched-mode documents are saved in an extended version of a format called *text/enriched*, which is defined as part of the MIME standard. This means that your documents are transportable (even through email) to many other systems. In the future other file formats may be supported as well.

Since Emacs adds some non-standard features to the format (colors and read-only regions), not all systems will be able to recreate all of the features of your document, but they will get as close as possible.

The MIME standard is defined in Internet RFC 1521; *text/enriched* is defined in RFC 1563. Details on obtaining these documents via FTP or email may be obtained by sending an email message to [rfc-info@isi.edu](mailto:rfc-info@isi.edu) with the message body:

help: ways\_to\_get\_rfcs

See also the newsgroup [comp.mail.mime](mailto:comp.mail.mime).

**CUSTOMIZATION**

The `fixed` and `excerpt` faces should be set to your liking.

User-preference variables: `default-justification`, `enriched-verbose`.

You can add annotations for your own text properties by making additions to `enriched-translations`. Note that the standard requires you to name your annotation starting with *"x-"* (as in *"x-read-only"*). Please send me any such additions that you think might be of general interest so that I can include them in the distribution.

**TO-DO LIST**

[Feel free to work on these and send me the results!]

+ Conform to updated text/enriched spec in RFC 1896.

+ Be smarter about fixing malformed files.

+ Make the indentation work more seamlessly and robustly:

```
+ Create<indent> an aggressive auto-fill function that will keep the
paragraph properly filled all the time, without slowing down
editing too much. Refill mode is a start at this, but needs

</indent></indent>      <indent>improvement.

+ Refill after yank. [Refill mode does that.]

+<indent> Make deleting a newline also delete the indentation following
it.</indent>

+ Never let point enter indentation??

+ Notice and re-fill when window changes widths (optionally).

+ Deal with the 'category' text-property in a smart way.

+ Interface w/ Gnus, VM, RMAIL. Maybe Info too? </indent>(Gnus 5.9 copes
with text/enriched incoming mail.)<indent>

+ Support more formats: RTF, HTML...

+ Use Emacs 21 display features.

</indent>

<x-bg-color><param>blue</param><x-color><param>white</param><bold>Final Notes:</bold></x-color></x-b
g-color>

<indent>This code and documentation is under development. Comments and
bug reports are welcome.</indent>

<bold><x-color><param>white</param><x-bg-color><param>blue</param>Boris Goldowsky</x-bg-color></x-co
lor><x-color><param>light blue</param> </x-color></bold><x-color><param>light blue</param><fixed><<b
oris@gnu.ai.mit.edu></fixed></x-color><x-color><param>blue</param>

</x-color><x-bg-color><param>blue</param><x-color><param>white</param>April 1995; updated August 199
7</x-color></x-bg-color>
```

Copyright (C) 1995, 1997, 2001, 2002, 2003, 2004, 2005, 2006, 2007  
Free Software Foundation, Inc.

COPYING PERMISSIONS:

This document is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA