

# Emacs Configuration

deepwaterooo

February 11, 2023

## Contents

<b>1</b>	<b>TODOs: tracing bugs of emacs 28.1</b>	<b>1</b>
<b>2</b>	<b>Updates</b>	<b>1</b>
2.1	configuration for pdf-tools packages . . . . .	1
2.2	invalid time specification: sr-speedbar on MacOS . . . . .	2
2.3	BUG statement and partial fix . . . . .	5
2.4	so far let it be this way . . . . .	6
<b>3</b>	<b>starting point</b>	<b>8</b>
<b>4</b>	<b>References</b>	<b>8</b>

## 1 TODOs: tracing bugs of emacs 28.1

## 2 Updates

### 2.1 configuration for pdf-tools packages

- pdf viewer noter <==> Skim bi-directional linking configuration on the way, most emacs work is done. Need to learn how to use them though.

```

Help
hhj@Heyans-MacBook-Pro:~/emacs.d/elpa/pdf-tools
~/emacs.d/elpa/pdfBK-tools/server/autobuild -i ~/emacs.d/elpa/pdfBK-tools/

Installing packages
Skipping package installation (already installed)

Configuring and compiling
./configure -q && make clean && make -s

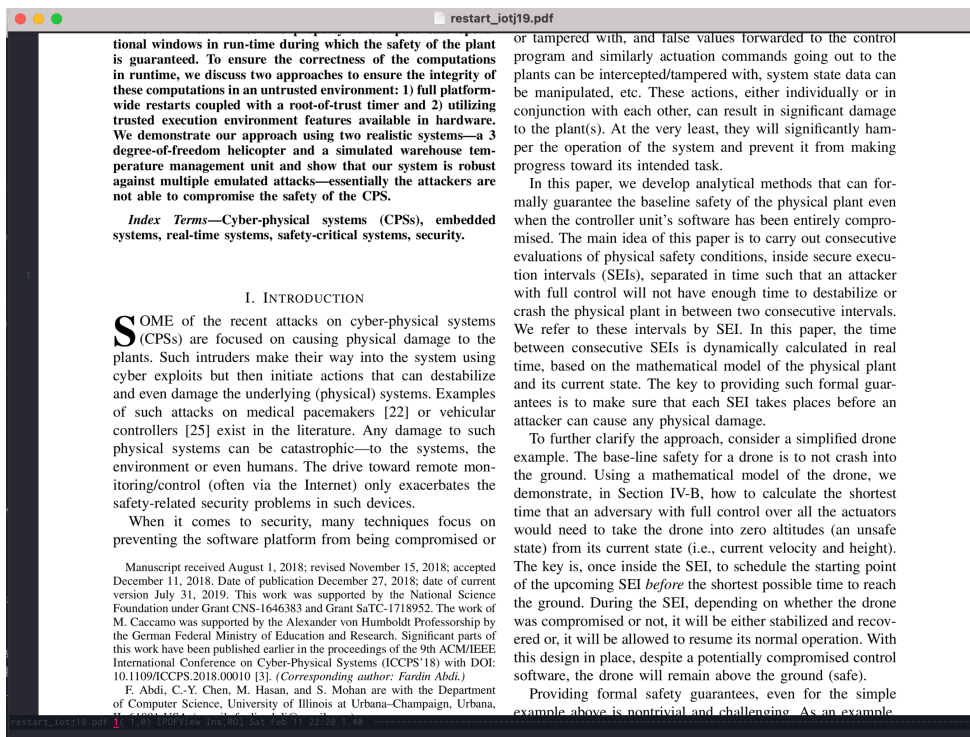
Is case-sensitive searching enabled ? yes
Is modifying text annotations enabled ? yes
Is modifying markup annotations enabled ? yes

test -z "epdinfo" || rm -f epdinfo
test -z "libsyntax.a" || rm -f libsyntax.a
rm -f *.o
epdinfo.c:354:19: warning: 'tempnam' is deprecated: This function is provided for compatibility reasons only. Due to security concerns inherent in the design of tempnam(3), it is highly recommended that you use mkstemp(3) instead. [-Wdeprecated-declarations]
    filename = tempnam(NULL, "epdinfo");
               ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/stdio.h:312:1: note: 'tempnam' has been explicitly marked deprecated here
__deprecated_msg("This function is provided for compatibility reasons only. Due to security concerns inherent in the design of tempnam(3), it is highly recommended that you use mkstemp(3) instead.")
^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h:215:48: note: expanded from macro '__deprecated_msg'
#define __deprecated_msg(msg) __attribute__((__deprecated__(msg)))
                                                ^
epdinfo.c:1548:5: warning: 'poppler_page_get_selection_region' is deprecated: Use poppler_page_get_selected_region instead [-Wdeprecated-declarations]
    poppler_page_get_selection_region (page, 1.0, selection_style, region);
    ^
/usr/local/Cellar/poppler/23.02.0/include/poppler/glib/poppler-page.h:75:134: note: 'poppler_page_get_selection_region' has been explicitly marked deprecated here
GList *poppler_page_get_selection_region(PopplerPage *page, gdouble scale, PopplerSelectionStyle style, PopplerRectangle *selection) G_GNUC_DEPRECATED_FOR(poppler_page_get_selected_region);
                                                                    ^
/usr/local/Cellar/glib/2.74.5/include/glib-2.0/glib/gmacros.h:694:18: note: expanded from macro 'G_GNUC_DEPRECATED_FOR'
#define __attribute__((deprecated("Use " #f " instead"))) \
               ^
2 warnings generated.

=====
Build succeeded. :0)
=====

```

- It has to be configured for M1. But I am still not getting any .tar file yet.



- could customer F5 toggle sr-speedbar, and make sis-mode work. But I do NOT really need



- Installed my emacs of version 28.1. But there is a bug of any verison emacs > 27.1, and I am NOT able to find a installable emacs 27.1 version any more.
- 不同电脑架构上可能因为架构的不同, 可以可能可以有某些优化. 又照一个单做了一遍, 似乎没有出错. <https://goykhman.ca/gene/blog/2022/2022-04-10-emacs-28.1-on-m1.html>
- 但是我没有没能加入那个补丁包. 暂时没能想好怎么加入那个补丁包. (**todo:** 改天可以尝试再把这个补丁包加进去)
- 因为构建是在原有现有的 mac 28.1.1 版本上构建的. 所以改动什么, 或是不曾改变, 又或者改不了不影响明显功能都是无从知晓的, 但是它最后的两个步骤的验证都是成功的, 应该还是构建成功了吧?

```

/usr/bin/install -c -m 644 etc/${docfile} "/usr/local/share/emacs/28.1/etc/${docfile}"; \
for installuser in $(LOGNAME) ${USERNAME} ${USER} `id -u` 2> /dev/null; do [ -n "${installuser}" ] && break ; done; installgrou
p=`id -g` 2>/dev/null && [ -n "$installgroup" ] && installuser=$installuser:$installgroup; \
chown ${installuser} "/usr/local/share/emacs/28.1/etc/${docfile}" || true ; \
else true; fi
Copying etc/DOC to /usr/local/share/emacs/28.1/etc ...
umask 022; ./build-aux/install-sh -c -d "/usr/local/bin"
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C lib-src install

Installing utilities run internally by Emacs.
umask 022 && ./build-aux/install-sh -c -d "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0"
exp_archlibdir=`cd "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0" && /bin/pwd` && \
if [ "$exp_archlibdir" != "/bin/pwd" ]; then \
for file in hexl movemail; do \
/usr/bin/install -c $file \
"/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0/$file" || exit; \
done ; \
fi
exp_archlibdir=`cd "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0" && /bin/pwd` && \
if [ "$exp_archlibdir" != "/bin/pwd" ]; then \
for file in rcs2log; do \
/usr/bin/install -c $file \
"/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0/$file" || exit; \
done ; \
fi

Installing utilities for users to run.
umask 022 && ./build-aux/install-sh -c -d "/usr/local/bin"
for file in etags ctags emacsclient ebrowse ; do \
/usr/bin/install -c $file \
"/usr/local/bin/" \
echo $(file) | sed -e 's/$/' -e 's,x,x,' \
|| exit; \
done
if test "/Users/hhj/selfSoft/emacs-mac/mac" != "`cd mac && /bin/pwd`"; then \
umask 022; ./build-aux/install-sh -c -d /Users/hhj/selfSoft/emacs-mac/mac/Emacs.app; \
(cd mac/Emacs.app; (tar -chf - . | \
(cd /Users/hhj/selfSoft/emacs-mac/mac/Emacs.app; umask 022; tar -xvf - \
&& cat > /dev/null))) || exit 1; \
fi
/usr/bin/install -c src/emacs "/usr/local/bin/" && echo emacs-28.1 | sed 's,x,x,'
/usr/bin/install -c -m 644 src/emacs.pdmp "/usr/local/libexec/emacs/28.1/aarch64-apple-darwin22.3.0"/emacs.pdmp
chmod 755 "/usr/local/bin/" && echo emacs-28.1 | sed 's,x,x,'
rm -f "/usr/local/bin/" && echo emacs | sed 's,x,x,'
cd "/usr/local/bin/" && ln -s "`echo emacs-28.1 | sed 's,x,x,'`" "`echo emacs | sed 's,x,x,'`"
/Applications/Xcode.app/Contents/Developer/usr/bin/make -C lib-src maybe-blessmail
make[1]: Nothing to be done for `maybe-blessmail'.

🍏 ~/5/emacs-mac on 📄 emacs-28.1-mac-9.0 73 em --version
GNU Emacs 28.1
Copyright (C) 2022 Free Software Foundation, Inc.
GNU Emacs comes with ABSOLUTELY NO WARRANTY.
You may redistribute copies of GNU Emacs
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING.

```

- 今天又尝试安装 Xcode 之后再构建一遍, 但是没有成功. 可能本身参考有些年代, 另外自己还完全不通这个部分, 所以暂时放一放. 改天有机会可以再回来研究一下, 错在哪里, 我如何才可能构建出自己的版本.

```

macronit.m:3482:15: warning: 'CTGetCoreTextVersion' is deprecated: first deprecated in macOS 11.0 - Use -[NSProcessInfo operatingSystemVersion] [-Wdeprecated-declarations]
    if (CTGetCoreTextVersion () >= kCTVersionNumber10_9)
    ^
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/System/Library/Frameworks/CoreText.framework/Headers/CoreText.h:61:10: note
: 'CTGetCoreTextVersion' has been explicitly marked deprecated here
uint32_t CTGetCoreTextVersion( void ) CT_DEPRECATED("Use -[NSProcessInfo operatingSystemVersion]", macos(10.5, 11.0), ios(3.2,
14.0), watchos(2.0, 7.0), tvos(9.0, 14.0));
^
1 warning generated.
CC      terminfo.o
CC      lastfile.o
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/charsets all
make[2]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/unidata charscript.el
make[2]: Nothing to be done for 'charscript.el'.
CLD      temacs
../build-aux/install-sh -c -d ../etc
/Library/Developer/CommandLineTools/usr/bin/make -C ../lisp update-subdirs
cp -f temacs bootstrap-emacs
rm -f bootstrap-emacs.pdmp
./temacs --batch -l loadup --temacs=pbootstrap
make[1]: *** [bootstrap-emacs.pdmp] Killed: 9
make: *** [src] Error 2

```

at 10:13:25 AM

```

/Library/Developer/CommandLineTools/usr/bin/make -C lib all
make[1]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C lib-src all
make[1]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C src VCSWITNESS='${srcdir}/../.git/logs/HEAD' all
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/charsets all
make[2]: Nothing to be done for 'all'.
/Library/Developer/CommandLineTools/usr/bin/make -C ../admin/unidata charscript.el
make[2]: Nothing to be done for 'charscript.el'.
rm -f bootstrap-emacs.pdmp
./temacs --batch -l loadup --temacs=pbootstrap
make[1]: *** [bootstrap-emacs.pdmp] Killed: 9
make: *** [src] Error 2

```

- 上面又成为一个需要改的东西：就是系统下如何从剪贴板自动生成写入文件 org-mode M-s
- 然后看见这里说可以自己构建一个，连 Xcode 也没有安装，就跑去构建了，当然不成功。这段时间太忙，XCode 要的空间太大了，暂时还不想。等改天有机会的时候倒是可以一试的
  - <https://stuff-things.net/2020/12/28/building-emacs-27-dot-1-on-macos-big-sur/>
- added key-bindings for opening from VSCode/Android Studio of current emacs buffer.
  - VSC Emacs can locate to each othr to correct row and col.
  - Android Studio could open current emacs buffer. but not to the row nor col.
- I liked recently configued Visual studio 2019 one-dark-pro theme, want to configue it for emacs, but ended up with any permission denied, renaming emacs initiating bug. reverted back for daily use, and may look into that bug for later reference.
- will reconfigure one-dark-pro theme later.
- fixed legency java-mode highlighting issue which I did not fix for years. Has been able to treat java-mode as java-mode Instead of using it as csharp-mode. Can not separate java-mode snippets from csharp-mode's.

## 2.3 BUG statement and partial fix

- in java-mode, the code style I expected is as followed:

```

class node {
    int v ;
    public node() {
        if (a > 0) // I don't want { } blocks when I have only one line statement inside blocks
        // a = 17;    // before fix:
            a = 17;    // now it can auto-indent
        b = 20;
        while (true) // same auto indents here
            j++;
    }
}

```

```

    }
}
class dklfjdj {} // <==== current un-auto-expanded version, bug right now for java-mode
class dklfjdj {
    | // <==== expected feature: once I typed '{', '}' will be autopaired(it does), but also auto-expand and cursor moves
}

```

- if while if while one line statement autoindent without {} fixed today for java-mode, but for kotlin-mode, this bug consists, make coders/programmers nuts.

```

fun getStringLength(obj: Any): Int? {
    if (obj is String)
        return obj.length // <==== BUG: need to fix auto-indent here for if else while etc without {}

    if (obj is String) {
        return obj.length
    }
    // 在离开类型检测分支后, `obj` 仍然是 `Any` 类型
    return null
}
fun dslfkj { // kotlin-mode, unlike java-mode, this feature works charming
    val a = 1720
}

```

- The {} can NOT auto-expand still bugs me a lot, I don't want to switch back to java-mode yet unless bug fixed and it auto-pands.
- java minor bug: Debugger entered-Lisp error: (void-function company-clear-completion-rules): this bug I will look into it recently, and expect it to be fixed so I could switch java-mode from csharp-mode as soon as possible.
- **csharp-mode** has been the one that works perfectly for these two features, {} auto expand, also if while one line statement autoindent without {}, so I used csharp-mode as java mode.

## 2.4 so far let it be this way

- Spent a whole day, half fixed the bug in the morning, but has broke java-mode completely and had to pull request from github once again.
- I actually cannot stay kotlin-mode insdie csharp's, look at following indent:

```

public fun thread(start: Boolean = true,
    isDaemon: Boolean = false,
    contextClassLoader: ClassLoader? = null,
    name: String? = null,
    priority: Int = -1,
    block: () -> Unit
): Thread {
    val thread = object : Thread() {
        public override fun run() {
            block()
        }
    }
    if (isDaemon) thread.isDaemon = true
        if (priority > 0) thread.priority = priority
            if (name != null) thread.name = name
                if (contextClassLoader != null) thread.contextClassLoader = contextClassLoader
                    if (start) thread.start()
                        return

```

- cause csharp-mode does not have ';' !!! so kotlin-mode stay as designed, I will have to bear the bug till I search and find good solutions.

- Recently I have used csharp-mode less, so temporarily let java-mode and kotlin-mode stay inside csharp's for a while. And I only want to practise kotlin to get a comfortable level referring github android projects. So once I get familiar, i still mainly using java-mode insides csharp's, which could be Ok, untill I get a job, and finally settle down to work on java-mode and `{|}` auto-expansion bug.
- today's
- will only update this repository when there is a need. emacs version 27.01 27.02
- working on leetcode interview questions, so have not configured any JDK IDE within emacs nor from csharp-mode, only take full advantage of emacs snippets for java algorithm problems coding. Uploaded so far some frequently used snippets for my own references.
- having not update this one for a while. pseudo-name .java java-mode inside csharp-mode so that I can skip some pairs of `{ }`-s
- java-mode csharp-mode file organiation scripts update
- Unhighlight leading or trailing whitespace
- org-move-tree make it slightly easier than before for manipulating small org files, and followed by integrating into one book file, and export into one pdf book. Seek for auto-updating integrated book file according to small chapter file updates later on when get spare time.
- fixed emacs org-mode export to pdf broken environment for personal laptop.
- configured company jedi environment for python3.
- Adding snippets for csharp-mode when debugging unity games.
- Remove not frequently used bothering commands from syslog-mode, and define simplified customized macro command for android SDK log analysis.
- **Enhanced syslog-mode**, with simplified textile-mode feature integrated for personal debugging log viewing propose. Will continuously improve relative features.
- logview-mode, log4j-mode, syslog-mode, in progress, so far only syslog-mode works, needs to combine textile-mode functions/hooks.
- textile-mode for android logs;
- sr-speedbar set fixed hight and width cater to current project file names length;
- fixed previously existing tab cannot indent line and region problem;
- company mode works convenient and as I expected;
- C-c f formating files according to needs. Fix minor bugs for java python csharp-mode swift-mode auto complete.
- clean auto-complete-mode, made repository more consistant.
- csharp-mode: fixed minor bugs for autopairing, as well as expand `{ }` for function scope.
- swift-mode using swift3
- org-mode src code highlight is on, just I forgot to specify language before.

- emacs key-bound for mac keyboard, so that it would be convenient for me to type some specific keys.
  - exchanged the position of `^Control` and Capslock;
  - exchanged the position of Option and Command keys;
  - through mac system preference.
  - I tried this yesterday, but after having used window's keyboard for all these years for emacs, it is still very difficult to get used to the mac keyboard even after key exchanges.
  - changed keyboard today actually so that I could type more conveniently.
- other major-modes, for example: **java-mode**, **csharp-mode**, which I would need to use pretty soon, is ready for use now (auto-complete + yasnipet etc).
- All the minor warnings, warning messages when starting emacs, modes fixes are all fixed, a clean Emacs open ready for work.

### 3 starting point

- It is a new computer, and I did try to git clone from my own repository to new laptop, but after fixed errors and tried, **I promise I do NOT and can NOT bare the out-dated emacs 22.X any more, I have to move on.** I have to install newer versions for my own later on convenience.
- Instead of configuring my own again, this time, I tried from some "big" person's repository and try to make it work on my laptop (fixing errors, installing necessary packages etc), as well as comment out some complicated modes and customization so that I would still be able to use and like my current emacs interface.
- It is the first time I tried from some big person's (or any person's configurature completely), it was tidious to fix all the errors at beginning (I spent more than 2 days on it last week. For me it just took too much time), but so far I like some of the features that had been annoying me before, but I have not and was not able to find good solutions to solve it, like how to auto-complete words when in scripte comment line or in quotes. I like these detailed features which I did try by writing my own snippets from yasnipet mode before.
- So far, org-mode is not perfect, but it is a fully functional one that I could use and help convenient a game developer's daily work.
- Will devote more time to understand emacs better, and to solve my own problems and make it more convenient for me to use when I need some specific features.

### 4 References

- <https://github.com/redguardtoo/emacs.d>