

## はじめに

NetBSD のパッケージシステム、pkgsrc。この使い方のメモ。

pkgsrc は NetBSD 以外でも使えるように配慮されている(たとえば、Solaris とか)けど、ここでは NetBSD 上のはなしのみ。ただし、他の OS で使っている人も、参考にはなるかも。

間違っていたり、もっとよい方法があったら、教えてね。

なお、バイナリーなパッケージの活用方法(つまり、pkgin を主に使う方法)はこの文書では取り扱わない。

## リンク

- [NetBSD/pkgsrc](#): 有名なところで、かなり情報があります。一度は見てみましょう。
- [pkgsrc.se](#): pkgsrc と wip のパッケージを検索できます。本家より使いやすいです。
- [The pkgsrc user's guide \(The pkgsrc guide Part I\)](#): 本家の解説文書。doc/pkgsrc.html と同じもの。一度はみてみませう。
- [pkgsrc 利用者向けの手引き](#): 上記の(ちょっと前の Ver. の)翻訳。こっちを見てから上を読むといいかも。
- [pkgsrc](#): NetBSD Wiki の tutorials より。同じく一度はみてみませう。

## 比較

### FreeBSD の ports とは違うところ

- portupgrade はない。
- portsnap はない。
- /etc/mk.conf が設定ファイル。
- /usr/pkg 以下にインストールされる。
- /usr/pkg/etc/rc.d にあるファイルでも自動起動しない。
- 起動スクリプトは、/usr/pkg/etc/rc.d や /etc/rc.d にインストールされない。
- /usr/pkg/share/example/rc.d にインストールされ、/etc/rc.d 以下には手動(=ユーザ自身)が配置する。
- FreeBSD では、「package(s) / パッケージ」という言葉は、port(s) をコンパイルして作った、バイナリーなパッケージの意味だが、pkgsrc では、pkgsrc に含まれる、個々のソフトのこと。pkgsrc から作られた、バイナリーなパッケージは、そのものズバリ「binary package(s) / バイナリーパッケージ」とか、「precompiled package(s)」という。

### Linux の各 distribution のパッケージシステムとは違うところ

- ソースから make してインストール、が基本。その際のソースを ftp/http で get -> パッチあて -> configure -> make -> make install を自動化するもの。(Gentoo の ebuild やソースが入っていない SRPM みたいな感じ?)
- Debian の dselect や aptitude みたいなコンソールな GUI はない。
  - pkgtools/pkg\_select というものがあるが、全然使われていない。
- Fedora core の yum みたいな x11 な GUI はない。
- apt-get upgrade で一発 OK、というようなものはない。
  - バイナリーパッケージの場合は、pkgtools/pkgin を使うのがデフォルトになってきた。pkgin full-upgrade でアップデートできる。(ただし、apt-get のように、全てのソフトをアップデートできるわけではない。)
  - パッケージをアップデートする場合は、pkgtools/pkg\_rolling-replace を使うのがいろいろと便利。
- apt-get install で一発 OK、というようなものはない。

- ・ 上にも書いたように、バイナリーパッケージの場合は、pkgtools/pkgin を使うのがデフォルトになってきた。pkgin install で、インストールできる。
- ・ 起動の on/off はコマンド (chkconfig や update-rc.d のような) ではなく、/etc/rc.conf を編集する。
- ・ /usr/pkg 以下にインストールされる。

## 共通

### pkgsrc 全体の設定

distfiles (ftp 等でとってきたソース) の保存 dir などの、pkgsrc 全体の設定については、mk/defaults/mk.conf にデフォルト値の設定がある。(一度読んでみるといいと思う。変更したい場合は、/etc/mk.conf に書く。mk/defaults/mk.conf は編集しない。

### 各パッケージの設定

パッケージ個別の設定 (configure オプションとか) については、[pkgsrc-2005Q2](#) に合わせるかのようになり、PKG\_OPTIONS.\* 変数を使う方法になったみたい。

それぞれ固有の変数に入れるもの

Makefile をみて、固有の変数が何かをみて、/etc/mk.conf に

```
HOGE_USE_HUGA= yes
```

とか書くもの。

デフォルト(既定)値は、mk/defaults/mk.conf に書いてあるが、稀に Makefile にしかないものがある。

これに該当するパッケージはもうほとんどないはず。

PKG\_OPTIONS. パッケージ名という変数名に入れるもの

こちらが今後の主流みたい。

make 時にオプションについて、注意される。変えたいときは、注意されたとおり、make を止めて、/etc/mk.conf に

```
PKG_OPTIONS.hoge= hoge-aaa hoge-bbb
```

とか書く。

デフォルト値は、options.mk ファイルがある場合 (例:[mail/qmail](#)) はそれ、ない場合は Makefile に書いてある。

流れとしては、options.mk を作るという方向みたい。

たんじゅんに、オプション一覧をみたいのなら、

```
cd /usr/pkgsrc/category/hoge
make show-options
```

で OK。

自分で Makefile を編集 :)

pkgsrc 側でオプションが用意されていない configure オプションとかは、自分で書き加えるしかないよね。

/etc/mk.conf にオプションを書きたくない

変数で指定すればよい。make の引数でもいいし、

```
make PKG_OPTIONS.huga=huga-aaa
```

環境変数でもよい。

```
env PKG_OPTIONS.huga=huga-aaa make
```

起動ファイルが /usr/pkg/etc/rc.d にインストールされない

昔は上記 dir にインストールされていた。(今でもままだ。) 現在は、

```
/usr/pkg/share/examples/rc.d
```

にインストールされる。昔みたいに /usr/pkg/etc/rc.d にもインストールしてほしい時は、

```
RCD_SCRIPTS_DIR= ${PKG_SYSCONFBASE}/rc.d  
PKG_RCD_SCRIPTS= YES
```

を /etc/mk.conf に書いておく。

起動ファイルは /etc/rc.d に、設定ファイルは /etc 以下にインストールしてほしい

```
PKG_SYSCONFBASE= /etc  
PKG_RCD_SCRIPTS= YES
```

を /etc/mk.conf に書いておく。起動ファイルは手動でインストールするなら、もちろん

```
PKG_SYSCONFBASE= /etc
```

だけで OK。

/etc/rc.d/hoge start とやっても起動しない

起動スクリプトは、普通は /etc/rc.conf で起動を許可しない限り、起動しない。www/apache だったら、

```
apache=yes
```

と書く。

一時的な起動の場合は、たいてい

```
/etc/rc.d/hoge onestart
```

で OK。これだと、`/etc/mk.conf/etc/rc.conf` に書いてなくてもよい。

root ユーザ以外で、`make install` すると、root パスワードを要求されるかわりに `sudo` を使いたい

```
.if exists(${LOCALBASE}/bin/sudo)
SU_CMD= ${LOCALBASE}/bin/sudo /bin/sh -c
.endif
```

を、`/etc/mk.conf` に書く。( `sudo` をインストールしていない場合は、`security/sudo` を先にインストールして、必要な設定をしてから。)

A というホストにインストールしてあるパッケージと同じパッケージ群を、B というホストにインストールしたい

A に `pkgtools/pkg_chk` をインストールして、

```
pkg_chk -g
```

してできた `/usr/pkgsrc/pkgchk.conf` を、B にもって行って、B で `pkgtools/pkg_chk` をインストールした上で、

```
pkg_chk -a -C /path/to/pkgchk.conf
```

でいい(はず)。

自動でオリジナルのパッチをあてたい

たとえば、

```
LOCALPATCHES= /usr/local/pkgsrcpatches
```

を `/etc/mk.conf` に書いて、`mail/qmail` を `make` するとすると、`make patch` の直後に `/usr/local/pkgsrcpatches/mail/qmail` 以下のパッチがあたる。つまり、`mail/qmail/patches` 以下のパッチの後に `LOCALPATCHES` のパッチがあたるようになる。

`patch` コマンドの実行ディレクトリは、各パッケージの `patches` と同じく、`WRKSRC` になるので、そのようにパッチをつくっておかないと `make` の途中で止まって悲しい思いをすることになる。:) ( `WRKSRC` って? という人は、[The pkgsrc developer's guide](#) を読もう。)

ちなみに、パッチファイルの名前はなんでもよい。ls の順番であたる。

パッケージの依存関係をみたい

`Makefile` を読むという手もあるけど、

```
make show-depends
```

で OK。既にインストールしているパッケージなら、

```
pkg_info パッケージ名
```

でも OK。

distfiles をとってくるときに、近くのミラーサイトをつかいたい

mk/defaults/mk.conf の下のほうに、各国の例が載っているので、/etc/mk.conf にコピーして、コメントアウトを外せば OK。(もちろん、自分で書き換えてもよい。大学内からだと、jaist にしたり、DTI ならだと、ftp.dti.ne.jp にしたり、いじるよね。)

コンパイルする時に自動で適当な最適化をほどこしたい

devel/cpuflags をインストールして、man の EXAMPLES をみて、mk.conf に追加しよう。

ただし、i386 とか amd64 とか、メジャー (?) な port でないかぎり、うまく動かなくなることもあるような気がする。(その時は mk.conf から消せばいいだけだけど。)

cpu がちゃんと判断されるかどうかは、cpuflags というコマンドがインストールされているので、実行してみればわかる。

並列コンパイル (make -j 数字) したい

/etc/mk.conf に

```
MAKE_JOBS=2
```

とか書くとそうなる。(数字は並列数。)ただし、全てのパッケージでそうなるわけではない。(うまくコンパイルできないものもあるので。)

make clean で、依存関係のパッケージも clean してほしい

/etc/mk.conf に

```
CLEANDEPENDS=yes
```

と書くと、そうなる。

一時的にそうしたい場合は、make clean-depends が使えますよ。

もう使わない古い distfiles を消したい

pkgtools/lintpkgsrc をインストールして、

```
lintpkgsrc -or
```

すると、pkgsrc ディレクトリの中と比較して、消してくれる。

古いパッケージを全自動でアップデートしたい。

全パッケージのアップデートを全自動でやりたいのなら、pkgtools/pkg\_rolling-replace が使える。

詳しくは [[pkg\\_rolling-replace を使ってみよう](#)] 参照。

特定のパッケージと、それに依存するパッケージを全自動でアップデートしたい。

全自動でやりたいのなら、pkgtools/pkg\_rolling-replace が使える。詳しくは [[pkg\\_rolling-replace を使ってみよう](#)] 参照。

Fedora でいう alternatives コマンド ( Debian だと update-alternatives ) はないの？

pkgtools/pkg\_alternatives がそれ。引数とかは全然違うけど、考え方は同じようなもの。詳しくは man をみよう。(ただし、サポートしているパッケージはとても少ない。)

pkgsrc でクロスコンパイルはできるの？

NetBSD の特徴として、システムをクロスコンパイルする (たとえば、NetBSD/mac68k を NetBSD/amd64 上で作成する) ことはとても簡単にできるんだけど、pkgsrc については、今のところ NetBSD 1.\* 系で、かつ crosspkgtools/crosspkg-netbsd1-\* にあるものを除いては、できません。ただし、distcc をつかってクロスコンパイルする方法があるようです。

doc/[HOWTO-distcc](#) を参照のこと。

(注: doc/[HOWTO-use-crosscompile](#) という文書が 2013 年 5 月に追加されてて、今後は楽しみではあります。)

デフォルトのオプションを消したい

```
PKG_OPTIONS.hoge= -hoge-aaa
```

などと /etc/mk.conf に書く。デフォルトのオプションから、hoge-aaa オプションだけははずすことができる。

たくさんのパッケージに共通のオプションを書きたい

```
PKG_DEFAULT_OPTIONS= -arts -canna sj3
```

などと /etc/mk.conf に書くとよい。

anonymous cvs でアップデートしたら、japanese とか nessus-libraries とかが pkgsrc dir の直下に見える

anonymous cvs のアップデートを途中で止めるとそうなる。やり直しましょう。

インストールパッケージの脆弱性情報を知りたい

その昔は security/audit-packages というパッケージをインストールしていたんだけど、今は、pkgtools/pkg\_install に統合されていて、NetBSD 5.0 以降は、base system にも入っている。( /usr/sbin/{audit-packages,download-vulnerability-list} として、見つかるはず。)

audit-packages が、脆弱性のあるパッケージを表示するコマンド。download-vulnerability-list が、脆弱性データベースファイルをダウンロードしてくるコマンド。( ... だったのだが、今

( pkg\_install-20090201 以降 ) は、 /usr/sbin/pkg\_admin に統合されて、 /usr/sbin/{audit-packages,download-vulnerability-list} は、ラッパーのシェルスクリプトになった。)

まず、 security/audit-packages をインストールしている人は、アンインストールした上で、 pkgtools/pkg\_install をインストール/アップデートしよう。( でないと、脆弱性情報を取得できなくなっているはず。 )

pkg\_install パッケージは時々アップデートされているので、 Release のまま使っているひとは、 Stable をおっかけるか、 pkgtools/pkg\_install をインストール/アップデートした方がよい。

pkgtools/pkg\_install をインストールしたら、 メッセージ が出るので、よく読もう。( 既にインストールしている人は、 pkg\_info pkg\_install で読める。 )

make なんとかって、 いったい何が書けるの？

旧 NetBSD Wiki にあった The pkgsrc make target reference が、よくまとまっていて、便利です。 NetBSD 公式 Wiki にある pkgsrc targets がざっとまとめていて便利です。

make install っても、 バイナリーパッケージができてたのに、できなくなっただけ

以前は、 USE\_DESTDIR=yes していた設定が、 2011/4 からデフォルトになった。簡単にいえば、 バイナリーパッケージをいったん作ってから、インストールするようになった。これは現在も変わっていないので、 バイナリーパッケージはできている。具体的には、 work/packages の下にできる。

ところが、いつしか、 PACKAGES ディレクトリ ( デフォルトでは、 /usr/pkgsrc/packages の下 ) にコピーされなくなった。昔のように、 make install 時にできたバイナリーパッケージをコピーされるようにするには、 /etc/mk.conf に

```
DEPENDS_TARGET= package-install
```

を書く。

work ディレクトリがあちこちに残っているんだけど、一括で消したい

```
cd /usr/pkgsrc
rm -rf */*/work
```

では、量が多い時に困る。こういう時は、 pkgtools/pkgclean をインストールして使うと速い。

```
pkgclean
```

とすだけ。

anonymous cvs なんてイマドキじゃない。 git 使えないの？

非公式として、 <https://github.com/jsonn/pkgsrc> が今まであったんだけど、 <https://github.com/NetBSD/pkgsrc> が公式 github mirror としてできました。

NetBSD の base system に入っているものは使わずに、pkgsrc に入っているものを使いたい

Postfix の場合は、[NetBSD で Postfix](#) 参照。

以下、net/ntp4 を例にあげる。

まず、インストール。ここまでは、なんら問題ないはず。

もし、ライブラリ群まで pkgsrc のものを使いたい場合(たとえば、net/ntp4 だと、openssl はそのままだと、base system のを使うけど、security/openssl を使いたい場合) `PREFER_PKGSRC=yes` をして、`make install` するとよい。

つぎに、どうやって起動させるかを考える。通常、`/etc/rc.conf` に、`ntpd=YES` と書いても、base system のものが起動してしまう。これを、pkgsrc でインストールしたものを起動するようにするのは、`/etc/rc.d/ntpd` を `/usr/pkg/share/examples/rc.d/ntpd` で置き換えてしまうのが一番簡単で、よく行われる。

しかし、置き換えてしまうと、base system の更新の時に、配慮しないとイケないし、いちいち覚えておくのはめんどろ。そこで、[NetBSD で Postfix](#) で紹介したような、`mail/postfix/MESSAGE.NetBSD` にあるような方法をとる。ただし、もうちょっと手抜きをする。

まず、`/etc/rc.conf.d` を作り、起動スクリプトをコピーする。

```
cp /usr/pkg/share/examples/rc.d/ntpd /etc/rc.conf.d
```

つぎに、`rc.subr` の行と、`load_rc_config` の行と、`run_rc_command` の行をコメントアウトする。ntp4 だと、以下のような感じ。

```
--- /usr/pkg/share/examples/rc.d/ntpd 2012-02-26 23:05:17.000000000 +0900
+++ /etc/rc.conf.d/ntpd 2012-02-27 00:03:32.000000000 +0900
@@ -8,7 +8,7 @@
# BEFORE: LOGIN
# KEYWORD: chrootdir

- . /etc/rc.subr
+# . /etc/rc.subr

name="ntpd"
rcvar=$name
@@ -48,5 +48,5 @@
rc_flags="-u ntpd:ntpd -i ${ntpd_chrootdir} $rc_flags"
}

-load_rc_config $name
-run_rc_command "$1"
+# load_rc_config $name
+# run_rc_command "$1"
```

これで、OK。

## それぞれ

editors/vim で日本語が使いたい [2013/8/16]



昔の vim パッケージは、日本語が使えなかった。これは、

```
--enable-multibyte --enable-xim
```

が configure オプションになかったため。現在は、

```
--enable-multibyte
```

はつくようになったが、それ以外はつかない。Linux の各 distribution のパッケージにあるような vim にしたい時は、最低、

```
--enable-multibyte --enable-xim --enable-fontset --with-features=big
```

はほしい。editors/vim-share の Makefile.common をみると、VIM\_EXTRA\_OPTS に設定すればいいみたいなので、

```
VIM_EXTRA_OPTS= --enable-xim --enable-fontset --with-features=big
```

を設定すればよい。

さらに、デフォルトのままでは、OS 標準の iconv (Citrus iconv) を使うのだが、文字コードの判定ができない等、使いづらいので、GNU iconv を利用するように変更する。これは、USE\_GNU\_ICONV=yes を指定して、make すればよい。

(参考: "editors/vim の文字化けについて" と "editors/vim で autodetect が有効にならない")

x11/aterm で日本語を入力したい [2005/5/16]

(かきかけ。)

FreeBSD の ports にあるパッチをあてれば、xim で入力できるようになります。

GQmpeg で日本語を表示したい [2009/1/17]

デフォルトの skin の場合、ソースの中にフォント指定が

```
.*-helvetica-bold-r-.*-14-.*-.*-.*-.*
```

のように埋め込まれているので、そのままでは、mp3 ファイルの id タグの日本語表示ができない。

そこで、デフォルト skin を一度、Skin Editor を使って、いったんファイル群に保存する。GQmpeg を起動し、適当なところで右クリックすると、メニューが出るので、Skin Editor を起動し、Skin Path を適当なところ (例: ~/.gqmpeg/skins/default) に保存する。

いちいち Skin Editor を起動して云々なんてやってられんという人は、この tar ball を ~/.gqmpeg/skins に展開するといいかも。

あとは、skindata\* の各ファイルをテキストエディタで開き、フォント指定をいじる。

そして、

```
gqmpeg --skin:default
```

で起動すると、日本語タグも OK。

しかし、環境設定で、Enable menu skins を On にしたときのメニューの文字化けはなおせないので、こればかりは、ソースをいじらないと直せない(と思う)。

ええと、インストールするときに、PKG\_OPTIONS.gqmpeg は gqmpeg-japanese は設定しているよね、もちろん。

「まだ、文字化けする！」という人は、たぶんタグの文字コードの問題。EUC-JP でない限り、ちゃんとでないよね。Windows からもってきたファイルだと、Shift\_JIS や UTF-8 だったりするので、それは残念ながら gqmpeg では convert して見るできない。

たんに日本語が表示できるプレーヤがほしただけだったら、audio/audacious を試してみよう。

audio/audacious で日本語を表示したい [2009/1/17]

プレイリストは日本語が出るのに、プレーヤのほうが出ない場合、設定 -> 外観の、「可能ならばビットマップフォントを使用」のチェックボックスをはずす。

inputmethod/anthy の anthy-dic-tool で辞書登録に失敗する [2005/7/2]

[pkgsrc-wip-discuss 52](#) と [tech-pkg-ja 3119] に投げた話。

結論としては、anthy-dic-tool のバグで、開発版 anthy では修正された。次の安定版がリリースされた際には pkgsrc に入ると思うので、それまでは、pkgsrc 版にも取り込まれた。ゆえに、6300nb1 以上にアップデートすればよい。

山城@ ofug さんの尽力に感謝。\_o\_

games/wesnoth をコンパイルするとエラーがでる [2005/8/17]

0.9.4 以上を make すると、

```
debase=`echo reports.o | sed 's|[^/]*$|.deps/&|;s|%.o$||'`; if c++ -DHAVE_CONF
reports.cpp: In function `reports::report
  reports::generate_report(reports::TYPE, const gamemap&, const unit_map&,
  const std::vector<team, std::allocator<team> >&, const team&, int, int,
  const gamemap::location&, const gamemap::location&, const gamestatus&, const
  std::set<std::string, std::less<std::string>, std::allocator<std::string>
  >&)' :
reports.cpp:422: internal compiler error: Segmentation fault
Please submit a full bug report,
with preprocessed source if appropriate.
See <URL:http://www.netbsd.org/Misc/send-pr.html> for instructions.
*** Error code 1
```

```
Stop.  
make: stopped in /opt/pkgsrc/games/wesnoth/work.i386/wesnoth-0.9.3/src  
*** Error code 1
```

というエラーがでるかもしれない。

悲しいかな、[tech-pkg-ja 3201](#) によれば、実メモリが 128MB では上記エラーがでるとのこと。まさにわたたくしは 128MB です。えーん。

ということで、swap を増やしても、どないしても make できないときで、大物っぽいときは、実メモリ不足を疑う、ということが必要かも。

だめもとで、reports.o の時だけ、コンパイル時の最適化を -O2 から -O に下げてみると、とおった。らっきー、ということで、これでしのぐことにする。

平塚さんに感謝。\_o\_

追記：実は、1.0 では make がとおるようになったんだけど、また似たようなことが起こった時のために、消さずに置いておくことにする。

## Java のソフトで日本語が文字化けする [2009/11/3]

[netbsd,09150] で書いた話。

lang/sun-jre15 と fonts/kochi-ttf をインストールして、java/sun-1.5/lib/fonts/fallback という dir をほって、kochi-\*.ttf を中にコピーすれば、とりあえず日本語が出るようになる。

[Supported Fonts](#) も参照のこと。

正攻法は、fontconfig.properties.src というファイルを書き換える方法らしい。

```
filename.Sazanami_Gothic=/usr/pkg/lib/X11/fonts/TTF/sazanami-gothic.ttf  
filename.Sazanami_Mincho=/usr/pkg/lib/X11/fonts/TTF/sazanami-mincho.ttf
```

などとすれば、OK。

## www/firefox,www/firefox2 で java が動かない [2007/6/1]

lang/sun-jre15 をインストールして、

```
In -s /usr/pkg/java/sun-1.5 /usr/java
```

するか、firefox の config(java.default\_java\_location\_others の値) をいじる。

あとは、[netbsd-java ML の 2004/4 の投稿](#) をみて、がんばるとできるらしいです。

lang/ruby を入れたんだけど、\*\* モジュールがインストールされてない [2005/7/31]

下記参照。

pkgsrc から ruby を入れたのに、/usr/pkg/bin/ruby がない [2011/10/23]

pkgsrc の ruby パッケージは、ruby 本家に含まれるいくつかの ruby モジュールが別パッケージになっている。

- lang/ruby1[89]-base: ruby の core の部分。
- lang/ruby1[89]: ruby1[89]-base + 別になっている ruby モジュール (iconv,gdbm,curses,readline,tk) がインストールされるメタパッケージ。
- lang/ruby: シンボリックリンクをはるメタパッケージ。

となっている。普通、lang/ruby を入れると、依存関係で、lang/ruby19-base がインストールされる。ゆえに、/usr/pkg/bin/ruby が無いひとは、

- lang/ruby19
- lang/ruby19-base

のうちどれかひとつをインストールしたので、たとえば、/usr/pkg/bin/ruby19 はあっても、/usr/pkg/bin/ruby は無い、ということになる。

ゆえに、lang/ruby をインストールすれば、/usr/pkg/bin/ruby19 から /usr/pkg/bin/ruby にシンボリックリンクがはられるので問題はなくなる。

(別解として、lang/ruby をインストールせずに、pkgtools/pkg\_alternatives をインストールする/使う、という手もある。結果的には同じことができる。)

なお、ruby のインストールのおすすめは、

- 1.lang/ruby19
- 2.lang/ruby

の順番で入れる。すると、なにも問題なく使えます。ただし、X 関係をインストールしていないサーバでは、ruby-tk を入れられてもうれしくないので、lang/ruby19/Makefile をいじって、その部分をコメントアウトしてから make install したほうがいいかもしれない。

すでに、pkgtools/pkg\_alternatives をインストールしている人は、lang/ruby をインストールできないので、その時は、lang/ruby19 をインストールした後で、

```
pkg_alternatives -gs rebuild
```

などとするとよいでしょう。( -gs の意味は、man を参照。)

graphics/mgl が make できない [2007/6/1]

最近の NetBSD/hpcmips だと、make できない。ホゲゆにによれば、gcc3 では make できないみたい。著者の [graphics/mgl に対するパッチ](#)によれば、

```
--- lib/emcons.c.orig 2005-01-17 05:09:53.000000000 +0900
+++ lib/emcons.c      2005-01-17 05:11:18.000000000 +0900
```



```

default:
-     if (size == romfont[4].height) {
+         ferom_font = &romfont[4];
+         s->_pen_font.opt = &romfont[4];
+         ferom_width = ferom_font->width;
+         ferom_height = ferom_font->height;
-     } else if (size == romfont[5].height) {
+         ferom_font = &romfont[5];
+         s->_pen_font.opt = &romfont[5];
+         ferom_width = ferom_font->width;
+         ferom_height = ferom_font->height;
-     } else {
+         ferom_font = &romfont[0];
+         s->_pen_font.opt = &romfont[0];
+         ferom_width = ferom_font->width;
+         ferom_height = ferom_font->height;
-     }
+ }

```

という修正が必要とのこと。(おおしまさんありがとう。)

また、[nbug:10248] によれば、src/sys/uvmm/uvmm\_extern.h に 2006 年秋に行われた変更によって、mgterm/mdate.c がダメで、

```

@@ -683,7 +684,11 @@
int
memmode()
{
+#if __NetBSD_Version__ > 105009900
+    struct uvmm_exp_sysctl uvmm_exp;
+#else
    struct uvmm_exp uvmm_exp;
+#endif
    int mib[2];
    int size = sizeof(uvmm_exp);
    int pagesize = getpagesize();

```

という変更が必要とのこと。(中治さん・おおしまさんありがとう。)

chat/navi2ch で bbsmenu がでない [2006/10/7]

現行の stable version である 1.7.5 では、そういう状態になる。navi2ch-1.75nb4 以上にアップデートすると、直っているので、アップデートしよう。

www/awstats で検索文字列などが文字化けする [2014/6/29]

AWStats の検索文字列の文字化けを解消の情報をもとに、手で修正してもよいが、ホスティングサービス総合研究所さんが 6.9 用パッチを公開されているので、make patch したあと、そのパッチを手パッチすれば、OK。

... だったのだが、www/awstats が、7.0 になって、上記のパッチはそのままではあたらない。手パッチしたあと、reject された部分は、手で編集しよう。

めんどくさい人むけの、パッチ。

Jcode モジュールをインストールするのを忘れないように！(converters/p5-Jcode だよ。)

pkgtools/pkg\_comp で ccache を使うと makeroot できない [2006/9/24]

なぜか、libkver のところで止まるという症状に悩まされている場合は、たぶん、REAL\_CCACHE 変数を指定している( = global ccache directory を使っている )はず。そういうひとは、普段から環境変数 CCACHE\_DIR を設定していると思う。pkg\_comp の場合は、CCACHE\_DIR は

/pkg\_comp/ccache にしなければいけないので、

```
env CCACHE_DIR=/pkg_comp/ccache pkg_comp makeroot
```

とすれば、うまくいくはず。パッケージのビルドも、

```
env CCACHE_DIR=/pkg_comp/ccache pkg_comp build misc/lv
```

などとしよう。

pkgsrc から lang/python[23]? を入れたのに、/usr/pkg/bin/python がない [2006/12/29]

pkgtools/pkg\_alternatives をインストールすると、できます。すでにインストールしている人は、

```
pkg_alternatives -gs rebuild
```

とするとできます。( -gs の意味は man を読んで下さい。)

isihara は、/usr/local/bin に必要なシンボリックリンクをはって、対処しています。

```
ln -sf /usr/pkg/bin/python24 /usr/local/bin/python
```

mail/wl など、Emacs 上で動くアプリケーションをインストールしようとする  
と、既に Emacs/XEmacs が入っているにもかかわらず、別のをインストールし  
ようとする [2011/4/3]

pkgsrc では、デフォルトの Emacs は mk/defaults/mk.conf に EMACS\_TYPE として書かれている。  
違うのを入れている場合は、mk.conf に EMACS\_TYPE を指定しておく。

```
EMACS_TYPE= emacs22nox
```

などと書けば OK。詳しくは、mk/defaults/mk.conf にある説明を読もう。

NetBSD/amd64 の multimedia/mplayer で Real Player の動画が再生できない  
[2007/4/8]

multimedia/realplayer-codecs は、NetBSD/amd64 では使えないので、悲しい思いをしていたのだが、  
下記の方法で再生できる。

要は、multimedia/mplayer/options.mk で明示的に disable されているのを止めて、mplayer 本家の  
essential-amd64 binary codecs をインストールすること。

まず、multimedia/mplayer-share/options.mk を下記のように変更する。

```
--- options.mk.orig      2007-01-14 02:33:33.000000000 +0900
+++ options.mk          2007-04-08 20:06:46.000000000 +0900
@@ -184,7 +184,7 @@
 CONFIGURE_ARGS+=
 DEPENDS+=
 .else
-CONFIGURE_ARGS+=      --disable-real
+#CONFIGURE_ARGS+=    --disable-real
 .endif
```

```
.if !empty(PKG_OPTIONS:Mplayer-runtime-cpudetection)
```

次に、`/usr/pkg/lib/codecs dir` をつくる。既にあれば、つぎへ。

`multimedia/mplayer` を既にインストールしているのなら、インストールし直す。configure のところで、

```
Checking for RealPlayer DLL ... yes (using /usr/pkg/lib/codecs)
```

と出てくるはず。

次に、mplayer 本家より、`essential-amd64 binary codecs` をとってきて、中身を `/usr/pkg/lib/codecs` に入れる。

これで使えるはず。詳細は、codecs のファイル名と `/usr/pkg/share/mplayer/codecs.conf` 参照。

### www/firefox で flash を見たい [2016/7/12]

その昔、`pkgsrc` には、flash プラグインは複数の Ver. があったが、現在は、一つしかない。

- `multimedia/adobe-flash-plugin11: 11.2.202.626`

ちなみに、linux のエミュレーションで動く。(つまり、`emulators/suse*` パッケージに依存している。)

上記は、`www/nspluginwrapper` を使うことで、プラグインとして動作する。(デフォルトでは、インストール時に、一緒にインストールされ、プラグインの設定も行われる。)この `nspluginwrapper` が、`multimedia/libflashsupport` に依存している。

`audio/pulseaudio` を使って、音声を出しているひとは、`PKG_OPTIONS.adobe-flash-plugin` に `pulseaudio` を指定しておくこと、よい。(インストール後の MESSAGE の設定を忘れずに。)

別の手段として、`multimedia/swfdec-mozilla` がある。ただし、開発は止まっているっぽいし、見ることができる flash は少ない。

別の手段はもうひとつあって、(代替として有名な) `multimedia/gnash` がある。こっちは、YouTube は見ることができるが、`ustream` やニコ動はだめっぽい。(でも、adobe 純正の plugin が動かないのであれば、こちらを使うしかないだろう。)こちらも、開発が止まっているようだ。

### NetBSD/hpcmips で www/w3m が make できない [2007/6/25]

`functable.c` と `tabtable.c` を作る `mktable` というプログラムが core を吐いて落ちてしまい、そこから先に進まない。どうも、`boehm-gc` が `hpcmips` ではうまく動かないらしい。

伊藤さんという方に、教えてもらった修正をして `boehm-gc` をインストールしなおし、そのあと `w3m` を make すると、できるようになった。



詳細は、モバイルギアで NetBSD を参照。

### modular-xorg でいろいろな X クライアントが make できない [2007/9/22]

デフォルトの XFree86 をインストールせずに、`/etc/mk.conf` に `X11_TYPE=modular` を指定して、modular-xorg 化しようとしているはず。

`XAW_TYPE` を `standard` (デフォルト値) 以外に指定している場合 (たとえば、`neXtaw` とか) にしていると、`x11/libXaw` がインストールされなかったり、インストールしていても、`buildlink3` で `include` されなかったりで、`Xaw` が入っていることを前提にしているパッケージは、のきなみ make できない (例 `editors/emacs`)。

いろいろひっかかるので、現状では、`/etc/mk.conf` には、すなおいに `XAW_TYPE` を指定せず、必要な時だけ、コマンドラインから指定した方がよいようだ。

### security/audit-packages がない [2008/1/14]

すでに `pkgtools/pkg_install` に統合されており、2008/1/13 に削除されたため。よって、`pkg_install` を `install/update` すればよい。(すでに `audit-packages` をインストールしている人は、`uninstall` してから。) cf. <http://www.jp.netbsd.org/ja/support/security/>

### ls を色つきにしたい [2008/1/20]

NetBSD の `/bin/ls` は色つきではない。Linux の `ls` に慣れている人は、色つきの方がうれしいかも。そんな人は、`sysutils/coreutils` (GNU `coreutils`) をインストールしてもいいけど、単に、色つきの `ls` がほしいだけ、の人は、`misc/colorls` か、`misc/gnuls` をインストールしよう。前者は、`ls` に `-G` オプションを追加して、色つきが使えるようにしたもの。後者は、`sysutils/coreutils` の `ls` だけを抜き出したもの。

### audio/tremor-tools で曲を再生後に core dump する [2008/1/21]

send-pr した話。audio/tremor-tools で ogg ファイルを再生すると、ちゃんと再生できるんだけど、最後に core dump する。あるいは、途中で `Ctrl + C` すると、core dump する。

どうも調べた限りでは、NetBSDの問題ではないようで、audio/vorbis-tools の方にはパッチがあたっているばかりだったので、それをまねて tremor-tools にも同じパッチをあてると、直ったようだ。

### www/apache{,2,22} で、ssl を有効にして起動したい。 [2008/10/20]

`rc.conf` に

```
apache=YES
apache_start=startssl
```

と書けばよい。(起動スクリプトを読むとわかる。イケてない。)

### misc/stellarium が起動しない [2009/1/29]

[nbug:11520] によれば、「X server に OpenGL 機能が必要、つまり modular Xorg 的には `glx extension` を有効にしておく必要がある」そうです。(おおしまさんありがとう。)

- ・ `x11/modular-xorg-server` を利用している場合、DRI を有効にして make しなおす。  
(`PKG_OPTIONS.modular-xorg-server= dri`。)

- xorg.conf or XF86Config で Load "glx" を書く。

また、(NetBSD/i386 5 系で、) intel ドライバを使っている場合で、必ず core dump する場合、

```
Option "DRI" "False"
```

すると、ちゃんと動くようになるかもしれません。(ただし、とても遅いです。)

emulator/gxemul で 4.0 以降の NetBSD/hpcmips を動かすと、時計が進まない [2009/7/21]

これは昔の gxemul 本家の Web サイトの News にも書いてあった不具合で、ずっと直っていなかったのだけれども、[netbsd,09744] で Tsutsui さんという方が原因を突き止めて下さって、回避策として、

```
とりあえず /etc/sysctl.conf に
sysctl -w kern.timecounter.hardware=clockinterrupt
とでも入れておけば時計は進むみたいです。
```

と教えて下さったとおりにすると、動いた。

また、[netbsd,09763] にあるとおり、pkgsrc が 0.4.7.2 に update され、修正パッチがあたるようになった。よって、上記の回避策をとるか、0.4.7.2 以降に update すればよい。

Tsutsui さんに感謝。\_o\_

audio/SDL\_mixer をつかうゲーム等で音が鳴らない [2009/11/3]

インストール時に出る、メッセージに気づいていないかも。

```
=====
$NetBSD: MESSAGE,v 1.5 2006/10/27 17:20:52 wiz Exp $

To make use of the MIDI capabilities in the SDL_mixer package, you
must install any of these packages:

audio/freepats
audio/eawpatches
audio/guspaches
=====
```

つまり、音源となるデータをインストールしないとイケない。

とにかく鳴ればいいのなら、audio/freepats をインストールしよう。(のこり二つは、サイズが大きい。)

インストールしても、まだ出ないのなら、timidity.cfg を作っていないのかも。audio/freepats では、インストール後に、

```
=====
$NetBSD: MESSAGE,v 1.2 2005/04/12 17:12:16 ben Exp $

To use timidity with freepats, you have a couple of options.

1) at the command-line:
```

```
timidity -L /usr/pkg/share/freepats -c timidity.cfg file.mid
2) in the system-wide timidity configuration:
cp /usr/pkg/share/freepats/timidity.cfg /usr/pkg/etc
```

=====

と案内が出る。2) をしておかないと、音が出ない。

games/onscripter も SDL\_mixer を使っているのですが、インストールしただけでは、音が出ません。:)

sysutils/munin-server をインストールしたが、文字がみんなトーフ (のような白い四角形) になる [2010/3/6]

sysutils/munin-server をインストールしたサーバは、NetBSD のインストール時に、xfont.tgz と xetc.tgz をインストールしなかったのでは。

インストールしたかどうかは、/etc/mtree 以下に set.xfont とか set.xsets とかいうファイルがなければ、たぶんインストールされていない。まずは、インストールしましょう。

インストールしても、まだトーフ、という人は、/etc/fonts/fonts.conf をみてみて。Font directory list は正しいところをみているかを確認しよう。(NetBSD 5 以降だと、/usr/X11R7/lib/X11/fonts が無いといけない。NetBSD を 4 以前から upgrade して使っている人ははまり道。)

gtk を使っているアプリケーション等のプルダウンメニュー等にもものすごく時間がかかる [2011/4/3]

root で、fc-cache コマンドを実行して、フォントキャッシュを作り直せば OK。

```
fc-cache -v
```

NetBSD を新規にインストールしたり、アップデートしたりした後、忘れずに実行しよう。(ちなみに、NetBSD 6 からは、インストール時起動時に実行されるようになるみたい。)

x11/mlterm で、xft を有効にできない or xft を無効にできない [2014/12/4]

その昔は、type\_engine = xcore がデフォルトだったので、xft 等が使えなかった。また、PKG\_OPTIONS.mlterm に xft2 を指定すると、今度は、type\_engine = xft しか使えないように、コンパイルされていた。

2014/11/2 に options.mk が変更され、デフォルトで、--with-type-engines=xcore,xft,cairo となった。mlterm-3.4.0 以降をインストールすれば、問題ない。

ruby をインストールしようとする、とても時間 (とメモリ) がかかる [2011/5/12]

付属ドキュメントを作成するのに、とても時間 (とメモリ) がかかるみたい。これは、PKG\_OPTIONS.ruby に -ruby-build-ri-db を指定すれば、それが省かれるようになる。

shell/zsh で、umount の引数の補完がエラーが出てうまくいかない [2011/7/28]

readlink(1) の引数が、Linux と違うので、うまく動かないっぽい。

とりあえず、エラーがでないようにするのは、make patch したあと、以下の修正を行えば直る。  
(もっとスマートな方法があるような気がするけど、よくわからない。)

```
--- Completion/Unix/Type/_canonical_paths.orig 2007-10-15 01:09:18.000000000 +0000
+++ Completion/Unix/Type/_canonical_paths
@@ -38,7 +38,7 @@ if (( $_opts[(1)-N] )); then
     files=($@)
     else
     for __index in $@; do
-     files+=$(readlink -qf $__index)
+     files+=$(readlink -f $__index)
     done
     fi

@@ -48,7 +48,7 @@ _canonical_paths_add_paths () {
     expref=${ origpref}
     [[ $origpref == (|*/). ]] && rltrim=.
     curpref=${${expref%$rltrim}:-./}
-     canpref=$(readlink -qf $curpref)
+     canpref=$(readlink -f $curpref)
     if [[ $? -eq 0 ]]; then
         [[ $curpref == */ && $canpref == *[^/] ]] && canpref+="/
         canpref+=$rltrim
```

pkgsrc で git をインストールして、github.com から、clone しようとする、エラーになる [2012/3/4]

たとえば、pkgsrc のリポジトリをとってこようすると、

```
% git clone --depth 1 https://github.com/jsonn/pkgsrc.git
Cloning into 'pkgsrc'...
error: SSL certificate problem, verify that the CA cert is OK. Details:
error:14090086:SSL routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed while accessing
https://github.com/jsonn/pkgsrc.git/info/refs
fatal: HTTP request failed
```

などとなって、SSL エラーになる。

これは、git がバックエンドに用いている curl コマンドがわかる root 証明書がないため。

まず、root 証明書を用意する。[curlの開発元が配布している](#)ので、それをとってこればよい。(信用できない場合は、同じサイトに、firefox のローカルインストールから、作成するスクリプトがあるので、それで作成すればよい。)

つぎに、用意した、cacert.pem を、git コマンドがわかるようにする。~/.gitconfig に以下のように書く。

```
[http]
sslCAinfo = /path/to/cacert.pem
```

これで、SSL エラーは出なくなる。

めんどくさいひとは、環境変数に、GIT\_SSL\_NO\_VERIFY=true を指定しておけば、SSL のチェックをしなくなるので、一時的につかう分にはこれでもよい。

```
% env GIT_SSL_NO_VERIFY=true git clone --depth 1 https://github.com/jsonn/pkgsrc.git
```

Tamago で Anthy を使いたい [2011/10/31]

editors/tamago は CVS 版なので、Anthy は使えるんだけど、不具合があったりする。白井秀行さんという方が『自家製 egg-anthy』という改良版をつくってみえて、それを藤原さんという方がパッケージにしたものが、[pkgsrc-wip](#) に [wip/tamago-anthy](#) としてあります。これを使うと幸せになります。

### textproc/py-sphinx で、make html 等すると、エラーになる [2012/1/21]

以下のようなエラーが出る。

```
Exception occurred:
  File "/usr/pkg/lib/python2.6/site-packages/docutils/utils.py", line 344, in decode_path
    path = path.decode(sys.getfilesystemencoding(), 'strict')
TypeError: decode() argument 1 must be string, not None
The full traceback has been saved in /tmp/sphinx-err-PfymiK.log, if you want to report the issue to
the developers.
Please also report this if it was a user error, so that a better error message can be provided next
time.
Either send bugs to the mailing list at <http://groups.google.com/group/sphinx-dev/>,
or report them in the tracker at <http://bitbucket.org/birkenfeld/sphinx/issues/>. Thanks!
*** Error code 1
```

これは、textproc/py-docutils の util.py がうまく動かないため。344 行目は

```
path = path.decode(sys.getfilesystemencoding(), 'strict')
```

となっており、(すくなくとも NetBSD 5.1 では、) sys.getfilesystemencoding() が、None を返しており、path.decode() は第一引数に None を許容しないので、上記のようなことがおこる。

とりあえず、以下の修正を行えば、問題を回避できる。

```
--- docutils/utils.py.orig      2011-07-07 06:49:19.000000000 +0000
+++ docutils/utils.py
@@ -349,6 +349,11 @@ def decode_path(path):
     path = path.decode('utf-8', 'strict')
     except UnicodeDecodeError:
         path = path.decode('ascii', 'replace')
+
+ except TypeError:
+     try:
+         path = path.decode('utf-8', 'strict')
+     except UnicodeDecodeError:
+         path = path.decode('ascii', 'replace')
+
return nodes.reprunicode(path)
```

### net/mikutter が起動しない [2012/1/27]

--debug オプションをつけて起動すると、文字コード関係のエラーがおきるときは、たとえば、

```
env LC_ALL=ja_JP.UTF-8 mikutter
```

とやって起動してみるとうまくいく場合がある。

じつは、/usr/pkg/bin/mikutter はただのラッパーの sh script なので、これに、

```
LC_ALL=ja_JP.UTF-8
export LC_ALL
```

とか書き加えちゃう、という手もある。

## notify-send (sysutils/libnotify) が動かない [2012/2/19]

sysutils/notification-daemon をインストールする。すると、

```
notify-send ほげ
```

で、動くはず。

なお、上記の例だと「ほげ」は日本語だと UTF-8 でないと動かない。EUC-JP 環境でコンソールを使っている人は、たとえば、converters/nkf をインストールして、

```
notify-send $(echo ほげ |nkf -w)
```

などしよう。

## www/{w3m,w3m-img} を migemo オプションを有効にしてインストールできない [2012/2/26]

pkgsrc の現在の ruby のデフォルトは、lang/ruby19。これを使おうとするが、textproc/migemo は、ruby18 依存なので、おかしくなる。回避策としては、w3m or w3m-img のインストールの時に、

```
make install RUBY_VER=18
```

とすれば、問題なくインストールできる。

## NetBSD/i386 6 系で multimedia/mplayer で flv が再生できない [2013/4/7]

mplayer で flv を再生しようとすると、

```
Unsupported PixelFormat 61
Unsupported PixelFormat 53
Unsupported PixelFormat 81
```

などといって、crash して、再生できない。

結論からいうと、[pkg/47132](#) と同じっぽく、書かれているとおり、

```
.if "${PKGPATH}" == "multimedia/mplayer"
  CFLAGS+= -mstackrealign -mpreferred-stack-boundary=4
.endif
```

と、/etc/mk.conf に書いて make install しておけば、再生できるようになる。

multimedia/mplayer 以外にも、multimedia/ffmpeg や multimedia/vlc も正常に動作しない場合は、同じ方法で直すようだ。

## net/nsd が起動しない [2014/6/29]

デフォルトのインストールのままでは、/var/nsd/nsd.db が作成できないため、起動しない。てっとりばやく NSD を参照のこと。

## ruby のパッケージがインストールできない [2016/2/25]

make はできるんだけど、make install しようとする、gem のエラーがでる場合は、一度、そのパッケージを make clean; make clean-depends; make distclean してから、再度試してみてください。

net/samba をインストールしていると、security/p5-GSSAPI がインストールできない [2017/1/12]

net/samba は security/mit-krb5 をインストールするが、pkgsrc デフォルトは、security/heimdal が Kerberos のデフォルトで、security/mit-krb5 のインストールを検知して、インストールに失敗する。

```
make KRB5_TYPE=mit-krb5 install
```

とすると、インストールできる。

devel/gobject-introspection が make できない。glib/gmacros.h が見つからないというエラーになる。 [2017/5/6]

結論だけいえば、[pkgsrc-users ML](#) によれば、/usr/pkg を symlink にしていると、このエラーになる。

null mount で代替する等で回避できる。

audio/abcde が動作しない [2017/5/6]

昔の abcde と違って、最新の abcde は、CDDDB のデータの代わりに、MusicBrainz を使おうとするんだけど、pkgsrc の abcde は Makefile に書かれているとおり、動作しないので、回避策として、今までどおり、CDDDB を使うよう、abcde.conf を

```
CDDBMETHOD="cddb"
```

と設定する。

次に、エンコードまではできるんだけど、タグが入らない、あるいは入るけど、文字化けしているときは、abcde コマンドの実行シェルの環境変数(たとえば、LANG 等)が、UTF-8 なものになっていないので、データが途中で壊れるのが原因。(EUC-JP 環境で生きている人(自分)等はひっかかる。)最も簡単な解決策は、abcde コマンド(これ自体はシェルスクリプト)に LC\_ALL=C を書いてしまうとよい。あと、細かなところでは、CDDDB から取得したタグ情報の編集エディタのパスが pkgsrc 前提になっていないので、そこらへんも abcde コマンド自体を書き換えるほかない。以下のような感じ。

```
$NetBSD$
--- abcde.orig 2017-01-18 13:56:14.000000000 +0000
+++ abcde
@@ -10,6 +10,7 @@
 # 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
+LC_ALL=C

VERSION='2.8.1'

@@ -2971,8 +2972,8 @@ do_cddbedit ()
 # If that fails, check for a vi
 elif which vi >/dev/null 2>&1; then
   vi "$CDDBDATA"
-   elif [ -x /usr/bin/vim ]; then
-     /usr/bin/vim "$CDDBDATA"
+   elif [ -x /usr/pkg/bin/vim ]; then
+     /usr/pkg/bin/vim "$CDDBDATA"
```

```

    elif [ -x /usr/bin/vi ]; then
        /usr/bin/vi "$CDDBDATA"
    elif [ -x /bin/vi ]; then
@@ -2980,13 +2981,13 @@ do_cddbedit ()
    # nano should be on all (modern, i.e., sarge) debian systems
    elif which nano >/dev/null 2>&1 ; then
        nano "$CDDBDATA"
-   elif [ -x /usr/bin/nano ]; then
-   /usr/bin/nano "$CDDBDATA"
+   elif [ -x /usr/pkg/bin/nano ]; then
+   /usr/pkg/bin/nano "$CDDBDATA"
    # mg should be on all OpenBSD systems
    elif which mg >/dev/null 2>&1 ; then
        mg "$CDDBDATA"
-   elif [ -x /usr/bin/mg ]; then
-   /usr/bin/mg "$CDDBDATA"
+   elif [ -x /usr/pkg/bin/mg ]; then
+   /usr/pkg/bin/mg "$CDDBDATA"
    # bomb out
    else
        log warning "no editor available. Check your EDITOR environment variable."

```

flac を作ろうとして、タグ付けに失敗する場合は、metaflac の入力 が UTF-8 前提になっているのが原因なので、abcde.conf で、

```
METAFLACOPTS=""
```

としておくとよい。

その他細かいこととしては、コマンド名等が違うので、以下のように、abcde.conf を設定している。

```

CDDA2WAV=cdda2wav
MD5SUM=md5
CDROM=/dev/rcd0d
MAXPROCS=2

```

www/firefox で uim で日本語入力できない [2017/5/6]

uim に gtk3 オプションを入れて make し直すと、入力できる。(デフォルトでは入っていない。)

NetBSD/amd64 で、www/firefox で Adobe Flash を再生はするものの音が鳴らない [2017/5/6]

この問題については、Tsutsui さんの、『[NetBSD + pkgsrc における oss, alsa, pulseaudio についてのメモ](#)』と『[NetBSD + pkgsrc + pulseaudio メモの補足](#)』に詳しく書かれているので、そちらを見て下さい。

自分が試した限りでは、必要なパッケージのインストール(option の変更なし)以外に、以下が必要だった。

- /usr/pkg/emul を /emul に null mount
  - symlink でもたぶんいいと思うけど、試してない
  - これをしておかないと、たとえば、/usr/pkg/emul/linux/usr/bin/aplay 等が動作しない

Tsutsui さんの上記文書によれば、libflashsupport パッケージ(Linux バイナリ)によって、(OSS に)音声を出しているようなので、これだけで動作するのだろう。

NetBSD/amd64 で、www/firefox52 で Youtube は再生はするものの音が鳴らない [2017/5/6]



www/firefox (Ver.53) の場合は、Firefox ごとクラッシュするようだ。

この問題については、Tsutsui さんの、『[NetBSD + pkgsrc における oss, alsa, pulseaudio についてのメモ](#)』と『[NetBSD + pkgsrc + pulseaudio メモの補足](#)』に詳しく書かれているので、そちらを見て下さい。

自分が試した限りでは、パッケージのインストール( option の変更なし)以外に、以下が必要だった。

- dbus を起動する
  - PC 起動時に自動起動するようにしておく。( /etc/rc.conf に dbus=YES の行を追加。)
- /usr/pkg/emul/linux/asound.conf を /usr/pkg/etc/asound.conf にコピーし、faillback oss の行をコメントアウト
- /usr/pkg/share/alsa/alsa.conf の @hooks の「/etc/asound.conf」を「/usr/pkg/etc/asound.conf」とする

alsa-plugins-pulse パッケージのインストールを忘れないように。

これでとりあえず音が鳴るようになった。

/var/shm にできる一時ファイルの容量問題がおきる場合は、Tsutsui さんの上記文書に回避策が載っているので参考に。