

MacOS X WorkShop
— apt-rpm system on MacOS X —

KOBAYASHI Taizo

2005/07/15

for MacOS X 10.3 (Panther)

MacOS X での LaTeX や emacs 等の
理工系研究環境が整うにつれ、
Linux と切り替えて仕事をするのが鬱陶しくなってきました。
かと云って、複数の計算機を管理する事を考えると、

コンソールだけで管理を完結したい！
Vine Linux 等で慣れた apt-rpm で管理したい！！

と云った欲求が生じてきます。

Linux で仕事をし、apt-rpm での管理に慣れた研究者達に
MacOS X での快適な研究環境を提供するのが
このプロジェクトの目的です。

ただし、このプロジェクトの成果物を利用して不具合が生じても、
プロジェクトとしてもプロジェクトに関係する如何なる人間も
一切責任を負わないものとします。

また、バグ報告やパッケージングの要望は歓迎しますが
迅速な対応は期待しないでください。

と云うよりも、要望をお持ちでしたら、

是非、要望を実現した姉妹 apt-rpm tree を作ってください！！

最後に、

我々は企業のサポート窓口ではありません！

こちらで不具合を再現出来る程度の情報がバグ報告に無い限り、
返事も対応も無いと考えてください。

*Copyright ©2004-2005 KOBAYASHI Taizo
All rights reserved.*

目次

1	はじめに	5
1.1	何故 apt-rpm か？	5
1.2	プロジェクトのポリシー	8
1.3	派生プロジェクトの歓迎	8
1.4	連絡先とメンバー	9
1.5	ライセンス	9
2	姉妹 trees !!	10
3	MacOS X WorkShop をインストールする	11
3.1	インストールする前に…	11
3.2	ターミナルの設定	11
3.3	インストール	12
4	MacOS X WorkShop の使い方	15
4.1	apt の「いろは」	16
4.1.1	毎回最初に必ずすべき事	16
4.1.2	パッケージの探し方	17
4.1.3	パッケージのインストール	17
4.1.4	パッケージの削除	18
4.1.5	パッケージの更新	18
4.1.6	後片付け	19
4.2	rpm の「いろは」	19
4.2.1	パッケージの情報あれこれ	19
4.2.2	パッケージのインストールと更新	21
4.2.3	パッケージの削除	21
5	パッケージを開発する	22
5.1	設定ファイルの編集	22
5.2	spec file のタグ	23
5.3	rpm macro	23
5.4	その他	24
6	インストーラを作る	26
6.1	段取り	26
6.2	作業場所を作る	26
6.3	インストールするファイル類を用意する	27
6.4	インストールする手順を所定の各ファイルに記述する	27
6.5	インストーラを作成する	29
6.6	ディスクイメージを作成する	31

7	apt-rpm tree を作る	33
7.1	段取り	33
7.2	tree を置く場所を用意する	33
7.3	パッケージを置く	33
7.4	データベースを作る	34
7.5	apt-line を記述する	34
8	パッケージメモ	35
8.1	Emacs 関連	35
8.2	T _E X 関連	36
8.3	X11 関連	38
8.4	開発関連	38
8.5	System 関連	39
9	スクリーンショット	40
10	既知の問題点と注意点	45
11	過去の議論	46
11.1	10.3 ~ 10.3-6 公開迄	46
11.2	10.3-6 ~ 10.3-7 公開迄	48
11.3	10.3-7 ~ 10.3-8 公開迄	52
12	謝辞	54

目次

1	「PackageMaker-Description」	29
2	「PackageMaker-Files」	30
3	「PackageMaker-Resources」	30
4	「PackageMaker-Info」	31
5	「PackageMaker-Version」	32
6	apt-get で update packages を download 中。	40
7	apt-get でパッケージを更新中。	40
8	apt-get でパッケージの更新中に StuffIt Expander が呼び出されているところ。	41
9	アップデート終了。	41
10	CarbonEmacs で mew を立ち上げているところ。	42
11	CarbonEmacs で yatex を用いて LaTeX の文章を書き Mxdvi でプレビューしているところ。Mxdvi からそのまま印刷可能です。	42
12	kterm 上の vim で kinput2 を通して日本語の文章を書いているところ。ことえりパッチを適用してあります。	43
13	Vine Linux と同じ gnuplot+ パッチを適用してあります。	43
14	Vine Linux と同じパッチを適用してあります。Color PS を生成できます。PS ファイルをダブルクリックすれば大抵の場合 PDF に変換でき、プレビューで確認と印刷ができます。	44

1 はじめに

このプロジェクトの目的は、
Linux で TeX や emacs を用いて仕事をしている人が、
ストレス無く MacOS X 上に環境を移す事

と
大学や研究機関等の計算機管理者が、
MacOS X 上に独自の研究環境を簡単に築き管理する事
にあります。

念頭に置いている Linux ディストリビューションは、大学で支持を得ている **Vine Linux**¹ です。
この Vine Linux の中で日々の仕事に必要なと思われるパッケージを独自に構築し直した物が、
MacOS X WorkShop の実態です。

現在公開しているパッケージは MacOS X 10.3 (Panther) に対応した物で

このディレクトリ² 以下に置いてあります。

rpm2html による **RPM 解説データベース**³ もご利用ください。

MacOS X WorkShop 固有の拡張を施してあるパッケージの内容に関しては
「パッケージメモ (Section8 参照)」をご覧ください。

尚、これらのパッケージを作るにあたり、
藤井恵介さん⁴ が公開していらっしゃる MacOS X 用の rpm packages を手本に致しました。
ここに感謝致します。

1.1 何故 apt-rpm か？

楽ができるから です。

例えば、TeX の環境を構築したいのであれば、
ターミナルを開いて

```
$ sudo apt-get update
$ sudo apt-get install OSX-base
$ sudo apt-get install task-tetex
$ sudo apt-get clean
```

とすることで TeX 関連のパッケージをまとめてインストールし、
且つ、面倒な各種設定まで自動で片付けてくれます。

¹<http://www.vinelinux.org/>

²../Panther/

³rpm2html/

⁴<http://www-jlc.kek.jp/~fujiik/mklinux/Packages/rawhide/diary/index-j.html>

パッケージの更新はバグが見つかる度に成されますが、その場合でも、

```
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get clean
```

でおしまいです。

いちいちインストールし直す必要は無いのです。

どうですか？
かなり楽が出来そう
ではありませんか？

MacOS X 上での UNIX 研究環境を構築するには **Fink**⁵ をはじめ、琉球大学の **EasyPackage**⁶ や、Apple が標準に据える確率が高い **DarwinPorts**⁷ 等があり、各々がそれぞれのパッケージングシステムを持っています。他にもパッケージングシステムを持たない総合情報として **Mac Wiki**⁸ が在り、自分が必要とするソフトを手で一つ一つ入れる事も出来ます。

当然の事ですが、それぞれに利点と欠点があります。Fink の利点は何と言ってもパッケージの多さでしょう。その反面大きなプロジェクトである為に、我々の些細な（しかし研究上無視出来ない）変更を施すのは余分な労力を要します。また、少なくとも小林は Fink にコミットする余裕は全くありません。

EasyPackage の利点は琉球大学で既に利用されていて、且つ、ある程度活発なコミュニティが存在していることだと思います。しかし rpm の様に枯れた技術とは言い難く、Linux の使用歴が長い研究者にとっては多分に不安を覚えるのも事実です。

DarwinPorts は Apple に最も近い存在ですが、既にかかなり大きなプロジェクトになっている事と、小林が FreeBSD を殆ど知らない事から対象外になっています。

apt-rpm を用いる副次的な利点としては、同じシステムを用いている **Vine Linux 等 Linux の成果を活かし易い** 事があげられます。以下、システムの簡単な概要を説明します。

⁵<http://fink.sourceforge.net/>

⁶<http://www.ie.u-ryukyu.ac.jp/darwin2/>

⁷<http://darwinports.opendarwin.org/>

⁸<http://macwiki.sourceforge.jp/cgi-bin/wiki.cgi>

rpm⁹ は Red Hat¹⁰ が Linux distribution のパッケージングシステムとして開発したものです。rpm, rpmbuild 等のコマンドを通して、パッケージの

- 構築
- インストール
- 更新
- 除去

を行います。パッケージ間の依存関係の情報をパッケージ自身が持っているので、必要なライブラリを抜かしてインストールする様なミスを防ぐ事が出来ます。

Vine Linux¹¹ 等、多くの Linux Distributer がこのパッケージングシステムを採用しており、MacOS X WorkShop に移植する際にそれらを拝借出来ます。また、ソフトベンダーが Linux 向けの製品を提供する場合は殆ど rpm 形式が使われており、Linux での標準的なパッケージングシステムになっています。

apt¹² は Debian GNU/Linux¹³ が Linux distribution のパッケージ管理ユーティリティとして開発したものです。

Debian の特徴は rpm ではなく独自のパッケージシステムを利用している事と、8000 以上の膨大なパッケージ数を抱えている事です。パッケージングシステムは兎も角、この様な膨大なパッケージを利用する為には何らかの強力なパッケージ管理ユーティリティが必要です。apt はその目的を果たす為に開発されています。

apt-get, apt-cache 等のコマンドを通して、

- 現在利用可能なパッケージの情報を得る。
- 更新されたパッケージを自動でアップデートする。
- パッケージ間の依存関係を自動で調整して適切なインストールと削除をしてくれる。

等、一度利用したら手放せなくなる機能を提供してくれます。

apt-rpm¹⁴ は Conectiva Linux¹⁵ が Linux distribution のパッケージ管理ユーティリティとして Debian の apt を rpm に対応させたものです。

⁹<http://www.rpm.org/>

¹⁰<http://www.redhat.com/>

¹¹<http://www.vinelinux.org/>

¹²<http://www.debian.org/doc/manuals/apt-howto/>

¹³<http://www.jp.debian.org/>

¹⁴<https://moin.conectiva.com.br/AptRpm>

¹⁵<https://moin.conectiva.com.br/>

MacOS X WorkShop では Conectiva の apt-rpm を Vine Linux が日本語対応にした物を流用しています。

rpm や apt の利用方法は「MacOS X WorkShop の使い方 (Section4 参照)」を御覧下さい。

1.2 プロジェクトのポリシー

さて、この MacOS X WorkShop プロジェクトには、以下のポリシーがあります。

- 管理に手間を掛けない。
- パッケージ数は必要十分に留める。
- 自分たちに都合の良い設定やパッチを用いる。
- それぞれの大学や研究室での派生プロジェクトを立ち上げやすくする。

です。

管理者がたった一人でも PowerBook と一日の時間さえあれば、
一通り全パッケージのメンテナンスが出来るくらいの
小さなディストリビューションに出来るだけ留めます。

ですから例えば ImageMagic, gimp は Mac に付属してくる GraphicConverter でほぼ代用できるので、
tree には加えていません。

結局のところ**如何に手間ひまを掛けずに必要十分な事を好き勝手に
にするか**が本音です。

煩わしい計算機管理は出来るだけ楽に済まし、自分の本分にリソースを集中する環境を作るのが、
MacOS X WorkShop の目的でありポリシーでもあります。

1.3 派生プロジェクトの歓迎

ですので、立命館大学物理学教室で立ち上がったこのプロジェクトをひな形にして、
各大学や研究室で独自の拡張や変更を施した
姉妹 apt-rpm tree が作られる事を歓迎します。

インストーラの作り方は「インストーラを作る (Section6 参照)」を、
apt-rpm tree の作り方は「apt-rpm tree を作る (Section7 参照)」を、
それぞれ御覧下さい。

貴方がたに必要なパッケージだけを集めた add-on tree も全く同様に作成可能です。
まずは、この MacOS X WorkShop を母体にした貴方独自の apt-rpm add-on tree を
local disk に作る事から始められる事をお薦めします。

もしも、姉妹 apt-rpm tree を作られ{る, た}際には是非ご一報ください。
姉妹 trees !! (Section2 参照) のページで紹介するとともに、
MacOS X WorkShop の apt-line に追加します。

そしてお互いに楽をし合いましょう。

1.4 連絡先とメンバー

MacOS X WorkShop に関する議論や連絡は **Mac Wiki**¹⁶ を利用させて戴ける事になりました。Mac Wiki で議論すれば、情報が蓄積されていき多くの人にとって有益です。

この web page が更新されるまでの変更は Mac Wiki でアナウンスしますので出来るだけチェックするようにしてください。

また、バグ報告は**基本的に OSXWS 標準の環境に対してのもの**にしてください。

勿論、.emacs.my.el 等を改変して独自の拡張を施すのは一向に構いませんが、

その結果現れた不具合の場合は**必ず OSXWS に問題がある事を特定してから報告**してください。

また、**問題が解決した場合にも必ず報告して、言いつ放しにはしない**でください。

現在のメンバーです。(順不同)

小林泰三 立命館大学物理学教室池田研究室 PD、非常勤講師

瀬戸亮平 立命館大学物理学教室倉辻研究室 D3

坂田泰啓 ニコン株式会社 ('03 池田研マスター卒業生)

新山友暁 立命館大学物理学教室池田研究室 M1

1.5 ライセンス

収録しているパッケージのライセンスは、

パッケージに収録しているソフトウェアのライセンスに従います。

\$ rpm -qi hoge でパッケージ hoge のライセンスを確認出来ます。

また /usr/local/share/doc/hoge 以下にもライセンスに関するファイルが在ります。

インストーラのライセンスは GPLv2 以降に従うものとします。

インストーラに同梱されている ReadMe.rtf, License.rtf を参照して下さい。

¹⁶<http://macwiki.sourceforge.jp/cgi-bin/wiki.cgi>

2 姉妹 trees !!

繰り返しになりますが MacOS X WorkShop の目的は、
Linux で TeX や emacs を用いて仕事をしている人が、
ストレス無く MacOS X 上に環境を移す事

と
大学や研究機関等の計算機管理者が、
MacOS X 上に独自の研究環境を簡単に築き管理する事
です。

一口に「楽をする」と云っても計算機環境に求められるものも好みも千差万別です。

その様な状況で管理者とユーザーの双方が楽をする為には、
それぞれの環境に合わせた apt-rpm tree を構築するのが一等です。

apt-rpm tree を零から新たに作るのは結構な作業に成りますが、
この MacOS X WorkShop をひな形にすれば、数日で実現可能です。

例えば、

デフォルトのログイン環境や `.emacs.el` を変えたければ、
OSX-Preferences パッケージを弄るだけで済みますし、
emacs に `lisp file` を加えたければ、
emacs-lisps パッケージに加えればおしまいです。

このページでは、姉妹 apt-rpm trees の紹介をします。

全て、MacOS X WorkShop の apt-line (`/priveta/etc/apt/sources.list` 内に記述) に加えてあります。
貴方の求めているものに最も近い tree をご利用ください。

- MacOS X WorkShop¹⁷

立命館大学物理学教室で立ち上げられ利用されています。

¹⁷<http://www.bach-phys.ritsumei.ac.jp/OSXWS/>

3 MacOS X WorkShop をインストールする

このセクションでは MacOS X WorkShop をインストールする手順について説明しています。

3.1 インストールする前に…

警告！

MacOS X WorkShop のインストール環境は、素の MacOS X に下記のパッケージをインストールした状況を想定しています。Fink との共存は可能かもしれませんがプロジェクトとしては考慮していません。インストール直後は path の設定から /sw が外されるので、Fink はそのままでは使えなくなるでしょう。また、他の方々が配布されている CarbonEmacs, teTeX 等がインストールされている場合、予期しない結果になる可能性があります。

MacOS X WorkShop は MacOS X 上で emacs などの UNIX ツールを利用する為の環境を構築するものです。

従って、MacOS X の標準インストール環境に加えて

以下を MacOS X インストールディスクから予めインストールしておく必要があります。

尚、現時点では **MacOS X 10.3 (Panther) のみに対応**しています。

- X11 (Panther install Disk3)
kterm, gv, gnuplot, xgraph, yaplot... 等の X ソフトを利用するのに必要です。
/usr/X11R6/bin/Xquartz がインストールされていれば大丈夫です。
- Xcode [ここ](#)¹⁸ から最新版を入手してください。
Apple が提供している開発環境です。インストールする時に **X11 開発環境を必ず選択**してください。

また、必須ではありませんが、
環境によっては以下をインストールしておく必要があります。

- AdditionalFonts.pkg (Panther install Disk3)
CarbonEmacs で利用する等幅 ascii font である Ayuthaya.ttf をインストールする為です。
2004 年 10 月 9 日現在で Ayuthaya.ttf は必須ではなくなりました。
但し、emacs で 10, 12, 14 point 以外のフォントを利用する場合は Ayuthaya.ttf
が必要です。/Library/Fonts/Ayuthaya.ttf がインストールされていれば大丈夫です。

3.2 ターミナルの設定

次に apt-rpm を操作するターミナルの設定をします。最低限文字コード設定は必要です。

1. 先ず「アプリケーション」→「ユーティリティ」の中にある『ターミナル』をダブルクリックして起動します。

¹⁸<http://developer.apple.com/tools/xcode/>

2. メニューバーの「ターミナル」から「ウインドウ設定...」を選択し『ターミナルインスペクタ』フローティングパネルを表示させます。
3. 「シェル」と表示されている選択項目をプレスして『ディスプレイ』を選択します。
4. 「文字セットエンコーディング」で『日本語 (EUC)』を選択します。
5. 「ワイドグリフは2桁とカウントする」にチェックを付けます。
6. 好みに応じて「アンチエイリアス処理を行う」にチェックを付けます。
7. 好みに応じて「フォント設定...」でフォントのサイズを変更します。

3.3 インストール

MacOS X WorkShop を始めるには

MacOS X WorkShop start kit MacOSX-WS-10.3j.dmg¹⁹

をダウンロードしてインストールします。

(ソース一式は MacOSX-WS-10.3j.tar.bz2²⁰ として置いておきます。)

このインストーラは以下の処理を行います。

1. apt-rpm のインストール

apt-rpm を利用する為の核となるものです。

apt, popt, rpm, beecrypt, bzip2 package の中から必要な物を抜き出したものです。

2. rpm data base の構築

```
$ sudo rpm --initdb
```

を実行します。

3. OSX-system, OSX-X11 パッケージのインストール

MacOS X に存在するリソースを rpm に知らせるパッケージをインストールします。

/private/etc/{csh.login,profile,zprofile} が置き換えられ/usr/local/bin 等にパスを通します。

/private/etc/X11/xinit/xinitrc が置き換えられ、パスを設定し各ユーザーの .Xclients を有効にします。

¹⁹MacOSX-WS-10.3j.dmg

²⁰MacOSX-WS-10.3j.tar.bz2

尚、オリジナルファイルは.rpmmorig のサフィックスを付けて保存されます。

インストールされる設定ファイルは **OSX-system**²¹ にてご確認ください。

4. ユーザー用初期設定ファイル (dot files) のインストールと配布

OSX-Preferences package をインストールします。

また、各ユーザーに以下の設定ファイルを配布します。

既に存在する時はファイル名の末尾を .rpmold に変えて保存した上で配布されます。

- **.Xclients**
X11 を起動する時に実行されるスクリプトです。
kterm や kinput2, login window をここで起動しています。
- **.Xresources**
主に xterm, kterm の設定が記述されています。
- **.bashrc, .bash_profile**
bash の設定ファイルです。
OSX-Preferences package の更新に対応する為に個人用の記述は **.bashmyrc** の中に記述して下さい。
- **.cshrc, .tcshrc**
csh, tcsh の設定ファイルです。
.tcshrc は .cshrc へのシンボリックリンクです。
OSX-Preferences package の更新に対応する為に個人用の記述は **.cshmyrc** の中に記述して下さい。
- **.zshenv, .zshrc**
zsh の設定ファイルです。
OSX-Preferences package の更新に対応する為に個人用の記述は **.zshmyrc** の中に記述して下さい。
- **.emacs.d, .emacs.el**
emacs の設定ファイルです。
OSX-Preferences package の更新に対応する為に個人用の記述は **.emacs.my.el** の中に記述して下さい。
- **.mew.el**
emacs 上で mew を利用するにはこのファイルを各自編集します。
- **.vimrc**
vi の設定ファイルです。
- **.rpmmacros**
rpm package を構築する時は、
予め各自このファイルを編集しておく必要があります。
- **.signature**
メールの署名ファイルです。

²¹OSX-system/

- rpm
rpm package を構築する時の作業ディレクトリです。

インストールされる設定ファイルは **OSX-Preferences-10.3.tar.bz2**²² をダウンロードしてご確認ください。

注意！

各ドットファイルはピリオドから始まるため、Finder から直接見る事は出来ません。
展開後に terminal 上で cat コマンド等を利用して確認して下さい。

これで様々な rpm package をインストール出来る様になります。

「MacOS X WorkShop の使い方 (Section4 参照)」を参照して必要なパッケージをインストールして下さい。

²²OSX-Preferences-10.3.tar.bz2

4 MacOS X WorkShop の使い方

MacOS X starter kit のインストールが無事済んだならば、後は自分が利用するパッケージを apt で入れるだけです。

お使いの計算機が firewall の内側である場合に限り、`.bashmyrc` を編集して環境変数 `http_proxy` や `ftp_proxy` を適宜設定しておく必要が在ります。該当箇所がコメントアウトされていますので、お使いの環境に合わせて記述してください。

starter kit をインストールした後に先ずすべき事は OSX-base パッケージを apt でインストールする事です。

MacOS X WorkShop で必須の根幹パッケージをインストールしてくれます。計算機がインターネットに接続されている事を確認してターミナルから

```
$ sudo apt-get update
$ sudo apt-get install OSX-base
$ sudo apt-get clean
```

を実行してください。

これが済んだら基本的には自由に必要なパッケージをインストールして載いてかまいません。

CarbonEmacs の環境を手っ取り早く構築したい人は、計算機がインターネットに接続されている事を確認してターミナルから

```
$ sudo apt-get update
$ sudo apt-get install task-emacs
$ sudo apt-get clean
```

を実行してください。

これだけで Carbon Emacs を利用する為の基本的な環境が構築されます。

注意！

emacs は `/Applications/Emacs.app` です。

「牛のアイコン」をダブルクリックして起動して下さい。

terminal や kterm 等から `$ emacs hoge.txt` としても起動出来る様に alias を設定してありますが、

引数として取れるのはファイル名だけです。

また、emacs で $\text{T}_{\text{E}}\text{X}$ のファイルを作成すると、文字コードは日本語 EUC になります。

TeX の環境を構築したい人は

```
$ sudo apt-get update
$ sudo apt-get install task-tetex
$ sudo apt-get clean
```

を実行してください。

これだけで齋藤さんの OTF パッケージでヒラギノを利用できる TeX 環境が構築されます。

注意！

TeX を利用する環境として emacs + YaTeX を想定しています。

pTeX は S-JIS で make されています。

terminal や kterm 上で emacs で作成したファイルをコンパイルする時には **platex-euc コマンド** を使用して下さい。

apt と rpm を用いて細かな操作をする必要がある人は以下の記述が役に立つかもしれません。勿論 man コマンドを活用して下さいね。

4.1 apt の「いろは」

Synaptic²³ の様な apt の GUI Frontend も世の中には在りますが、これは GTK2 を必要とし、利用する為には関連するパッケージ群を用意せねばなりません。残念な事に、それら必要なパッケージのうち openjade を MacOS X で構築できず、暫くは用意する予定はありません。(勿論どなたかが作ってくだされば大歓迎です！)

そこで、ターミナルでの apt の利用方法を簡単に紹介します。

4.1.1 毎回最初に必ずすべき事

apt を利用するには何は兎もあれ**データベースの更新**をする必要があります。これをしないと現在の apt line²⁴ の状態を反映出来ず、無いはずのパッケージをインストールしようとしたりしてまともに働いてくれません。

ですから apt を弄る時は、必ず最初に

```
$ sudo apt-get update
```

する様に癖をつけてください。

²³<http://www.nongnu.org/synaptic/>

²⁴apt が利用するパッケージやその情報が置いてある場所の事

4.1.2 パッケージの探し方

例えば、現在利用できる emacs に関するパッケージを知りたいとしましょう。
その様な時は `apt-cache search` を利用します。
具体的には

```
$ apt-cache search emacs
Mule-UCS - MULE-UCS is a coding system and character code translator system.
apel - Emacs 用の 基礎的な関数を提供するライブラリ
emacs -GNU Emacs エディタ
emacs-lisps - Carbon Emacs 用の便利な Lisp ライブラリ集
emacsen-common - Common facilities for all emacsen.
flim - Emacsen 用の message に関する表現形式や符号化のためのライブラリです。
mew - Emacs でメールを読むためのインターフェース
mew-common - Emacs/XEmacs 用 Mew 両方で利用するファイル/プログラム
readline - A library for editing typed command lines.
semi - Emacsen 用の MIME の機能を提供するライブラリ
task-emacs - emacs バーチャルパッケージ
yatex - 野鳥 (YaTeX) - Yet Another TeX mode for Emacs
```

の様にすれば、emacs に関連したパッケージの一覧が得られます。

4.1.3 パッケージのインストール

`apt-cache` を用いてインストールしたいパッケージが見つかったら、
`apt-get install` を利用してインストールします。

例えば、`a2ps` をインストールする場合には

```
$ sudo apt-get install a2ps
パッケージリストを読みこんでいます... 完了
依存情報ツリーを作成しています... 完了
以下の追加パッケージがインストールされます:
  psutils
以下のパッケージが新たにインストールされます:
  a2ps psutils
アップグレード: 0 個, 新規インストール: 2 個, 削除: 0 個, 保留: 0 個
1658kB のアーカイブを取得する必要があります。
展開後に 4783kB のディスク容量が追加消費されます。
続行しますか? [Y/n]
取得:1 http://www.bach-phys.ritsumei.ac.jp Panther/ppc/main psutils 1.17-10.3tk1 [78.1kB]
取得:2 http://www.bach-phys.ritsumei.ac.jp Panther/ppc/main a2ps 4.13b-10.3tk1 [1580kB]
```

1658kB を 6s 秒で取得しました (256kB/s)
変更を適用しています...

```
Preparing... ##### [100%]  
  1:psutils ##### [ 50%]  
  2:a2ps ##### [100%]  
完了
```

の様になります。

パッケージ間の依存関係が解決されて、psutils が同時にインストールされているのが判ります。

4.1.4 パッケージの削除

いらなくなったパッケージを削除したい時にはどうすれば良いでしょう？
その様な時は `apt-get remove` を利用します。
例えば、psutils を削除したい場合

```
$ sudo apt-get remove psutils  
パッケージリストを読みこんでいます... 完了  
依存情報ツリーを作成しています... 完了  
以下のパッケージが削除されます:  
  a2ps psutils  
アップグレード: 0 個, 新規インストール: 0 個, 削除: 2 個, 保留: 0 個  
0B のアーカイブを取得する必要があります。  
展開後に 4783kB が解放されます。  
続行しますか? [Y/n]  
変更を適用しています...  
Preparing... ##### [100%]  
完了
```

の様になります。

ここで psutils に依存している a2ps が存在する場合、a2ps も削除するかどうか確認してきます。
ですから、あるパッケージを抜いてしまったが為に動かなくなるパッケージは、バグでない限りありません。

4.1.5 パッケージの更新

計算機のソフトにバグはつきものですし、機能が追加されてどんどん更新されていくものです。
MacOS X WorkShop でも、当然バグつぶしに因るパッケージのアップデートはしていきますし、
開発元が新版をリリースすれば出来る範囲で追随します。
即ち、パッケージはどんどん新しくなっていきます。

その様な新しいパッケージに自動で更新する方法があります。

一つ目は **依存関係を解決する時に、パッケージの削除が伴わないものだけ**を更新する方法で、
`apt-get upgrade` を利用します。

二つ目は **パッケージの削除を伴っても依存関係を解決して最新の状態にする**方法で、
`apt-get dist-upgrade` を利用します。

小林は開発に携わっている事もあり、常に `apt-get dist-upgrade` しています。

4.1.6 後片付け

`apt-get` で取得したパッケージは
`/private/var/local/cache/apt/archives/` 以下に置かれます。
これは、`apt-get clean` を実行しない限り、残り続けます。
必ず最後に実行しておきましょう。

```
$ sudo apt-get clean
```

4.2 rpm の「いろは」

パッケージをインストールしたり更新したりするのは `apt` に任せれば良いのですが、
パッケージそのものを相手にする場合は `rpm` コマンドを直接操作する他ありません。
ここでは普段小林がよく使うコマンドについて簡単に解説します。

尚、パッケージの作成方法については「[パッケージの開発 \(Section5\)](#)」をご覧ください。

4.2.1 パッケージの情報あれこれ

今インストールされているパッケージの情報を知りたいとします。
先ず、今インストールされている全てのパッケージを知るには `rpm -qa` を用います。
実際には `sort` にパイプして

```
$ rpm -qa | sort | less
```

としたり、

```
$ rpm -qa | grep devel | sort
```

として目的のパッケージを探します。

こうして調べたいパッケージを見つけたならば、
何時誰が作ったパッケージで何時インストールされたのか、
等の情報を得ることができます。
それには rpm -qi を用います。
例えば a2ps の情報であれば

```
$ rpm -qi a2ps
Name           : a2ps                      Relocations: (not relocateable)
Version        : 4.13b                     Vendor: (none)
Release        : 10.3tk1                   Build Date:  金  8/20 11:53:53 2004
Install Date:  日  8/22 16:21:11 2004      Build Host:  ww5pt176.local
Group          : Applications/Publishing    Source RPM:  a2ps-4.13b-10.3tk1.src.rpm
Size           : 4511931                    License: GPL
Signature      : (none)
Packager       : KOBAYASHI R. Taizo        <xxxxxxx@mac.com>
URL            : http://www.inf.enst.fr/~demaille/a2ps/
Summary        : テキストなどの Postscript へのフィルタ
Description    :
a2ps は優れた印刷能力をもった、テキストを PostScript へ変換するフィルタ
です。
これは、プログラム言語や文字コード (ISO Latins, Cyrillic, EUC-JP 等)、
用紙、(インタフェースに対して)NLSなどを広範囲にサポートしています。
いくつかのファイルを別のアプリケーションでフィルタリングさせる機能も持っ
ており、DVI や PostScript 等を全く同じインタフェースで区別することなく印
刷することができます。
```

として入手出来ます。

では、a2ps で一体どのようなファイルが何処にインストールされているのかを知るにはどうすれば良い
のでしょうか。
それには rpm -ql を用います。

```
$ rpm -ql a2ps
/private/etc/a2ps-site.cfg
/private/etc/a2ps.cfg
/usr/local/bin/a2ps
/usr/local/bin/card
/usr/local/bin/composeglyphs
.....
```

```
/usr/local/share/ogonkify/ptmri-o.ps
/usr/share/info/a2ps.info.gz
/usr/share/info/ogonkify.info.gz
/usr/share/info/regex.info.gz
```

最後に、このパッケージの履歴をみてみましょう。
それには以下の様にします。²⁵

```
$ rpm -q --changelog a2ps |less
```

4.2.2 パッケージのインストールと更新

ダウンロードしてきたパッケージをインストールする方法や、
自分で構築したパッケージをインストールする方法を述べます。

例えば、パッケージ hoge-1.23-10.3tk1.ppc.rpm をインストールするには

```
$ sudo rpm -ivh hoge-1.23-10.3tk1.ppc.rpm
```

或は

```
$ sudo rpm -Uvh hoge-1.23-10.3tk1.ppc.rpm
```

とします。

-i オプションはインストールを、
-U オプションは更新を意味しますが、
殆どの場合 -Uvh で済んでしまいます。

他に、--force や --nodeps 等のオプションがありますが、
パッケージの作成をしない限り、まず使う状況は無い筈です。

4.2.3 パッケージの削除

大抵は apt-get remove で事足りるのですが、
開発作業中にどうしても依存関係を破壊しても一時的に削除しなければならない場合は
rpm コマンドに頼る他ありません。

その様な時は

```
$ sudo rpm -e --nodeps hoge
```

とします。

²⁵rpm-4.2.1-10.3tk8 では日付の表示がおかしくなっています。

5 パッケージを開発する

ここでは MacOS X WorkShop のパッケージを開発する方法を述べます。
コマンドは rpm ではなく rpmbuild を使います。

MacOS X WorkShop に固有の事項について説明しますので、
一般的な rpm パッケージの作成方法は、
Vine Linux の **Making RPM**²⁶ や、
Momonga Linux の **Specfile-Guidance**²⁷ を参考にしてください。

亦、パッケージに固有の項目に関してはパッケージメモ (Section8 参照) を参照して下さい。
尚、

```
$ rpm -i hoge-1.0-10.3tk1.src.rpm
```

とすると、
spec file は ~/rpm/SPECS に、
source files は ~/rpm/SOURCES に、
それぞれ入ります。

apt tree に在る rpm source package を利用するのであれば

```
$ cd ~/rpm/SRPMS  
$ apt-get source hoge
```

とすると、
hoge の source package が ~/rpm/SRPMS にダウンロードされた後に
spec, source files を所定の位置に展開してくれます。

5.1 設定ファイルの編集

rpm のパッケージを作る前に、パッケージャの情報を ~/.rpmmacros に記述しておきます。
vi 等のエディタで ~/.rpmmacros の packager の項目に、
アルファベットで自分の名前とメールアドレスを以下の様に整えます。

```
_%topdir ${HOME}/rpm  
%packager KOBAYASHI Taizo <xxxxxxx@xxxx.xxx>  
  
%_tmppath %{_topdir}/temp
```

²⁶<http://www.vinelinux.org/MakingRPM/index.html>

²⁷<http://www.momonga-linux.org/docs/Specfile-Guidance/ja/index.html>

これで貴方が作るパッケージには貴方の名前とメールアドレスが刻まれます。

5.2 spec file のタグ

ここでは MacOS X WorkShop 固有のタグに関する方針を述べます。

Version, Release rpm のパッケージは

(Name)-(Version)-(Release)-(architecture).rpm

の形をしています。

(Name), (Version) はパッケージングするソフトに依存するので一意に決定されますが、

(Release) の付け方はディストリビューション毎に取り決めがあるのが普通です。

MacOS X WorkShop では

Panther 10.3tk(release number)

Tiger 10.4tk(release number)

と付ける事にします。

(architecture) は特に指定しなければ ppc になります。

スクリプトやドキュメントだけのパッケージでは **BuildArch: noarch** を指定すると noarch になります。

defattr %files セクションに、そのパッケージに含まれるファイルを書き込みますが、

それらのファイルのオーナーとグループを指定してやる必要があります。

それが %defattr タグです。

MacOS X WorkShop では、

%defattr(-, root, wheel)

を標準とします。

5.3 rpm macro

ここでは MacOS X WorkShop 固有のマクロについて述べます。

デフォルトのマクロは /usr/local/lib/rpm/macros に記述されているので、パッケージを作成する前に必ず一度は目を通しておいてください。

まず、マクロの内容が MacOS X WorkShop 固有のものを列挙します。

_prefix /usr/local

基本的に全てのバイナリーやライブラリ、ドキュメント等は /usr/local 以下にインストールします。

_var /private/var/local

_sysconfdir /private/etc

__libtoolize /usr/bin/glibtoolize, /usr/bin/glibtool を使います。
/usr/bin/libtool はライブラリを作る ar, ranlib の代替になる存在のようで、
gnu の libtool とは役割が異なります。

次に、MacOS X WorkShop のみに存在するマクロを列挙します。

```
#-----  
#  
# MacOS X macro  
%_cpmac          /Developer/Tools/CpMac  
%_mvmac          /Developer/Tools/MvMac  
%_fontsdirmac    /Library/Fonts  
%_docdirmac      /Users/Shared/Docs/{name}-{version}  
%_appdirmac      /Applications  
%_stuffit        open -a StuffIt\\ Expander.app  
%_Xcode          open -a /Developer/Applications/Xcode.app
```

リソースフォークを持つファイルを扱う時にこれらのマクロを利用します。

5.4 その他

ライブラリに関する問題

libintl, libiconv 等が読み込まれなかったり、
/usr/include/{glob,regex}.h 等が gnu のものとコンフリクトしたりする事があります。
大抵、ld が multiple definitions of symbol.... や
undefined symbols.... 等とメッセージを出し、
該当する関数名とライブラリ名を表示してくれるので、
出来るだけ MacOS X 側が用意しているライブラリやヘッダファイルを利用する様にします。

ただし、libxml2 は、MacOS X 10.3.5 + Xcode 1.5 の場合、
ヘッダファイルの version が 2.5.4 で実際にインストールされているライブラリは 2.6.7 になっています。

このような場合は、そのライブラリを使うのを諦めるか、
新しい version のもので置き換えるかする必要があります。

MacOS X 本体にはあまり手を加えたくないなので、現状では諦めています。

libtool, autotools に関する問題

MacOS X 10.3.5 + Xcode 1.5 では、

libtool	1.5 (glibtool, glibtoolize)
aclocal, automake	1.6
autoheader, autoconf	2.57

が用意されています。

MacOS X WorkShop では、automake1.{478}, autoconf2.59 を用意しています。

これらは必要に応じて、aclocal-1.8 -I /usr/share/aclocal 等として利用します。

libtool を利用する時は、

configure の前で glibtoolize --copy --force とし、

configure の後で cp -f /usr/bin/glibtool libtool

とするとうまく行く事があります。

c++ に関する問題

MacOS X 10.3.5 + Xcode 1.5 の gcc3.3 では、

c++ の `#pragma interface/implementation` は使えません。

ソースにこれらがある場合、コメントアウトしておきます。

ですがこのままでは巨大なバイナリーが出来てしまうので、出来たバイナリーを strip してやります。

6 インストーラを作る

ここは MacOS X WorkShop 10.3 のインストーラを Panther 上の Xcode-1.5 で小林が開発した際の備忘録です。

屹度間違いがあるとおもいます。

総合的且つ正確な情報は Apple の **Tools: Software Distribution**²⁸ をご覧下さい。

インストーラのソース一式は「インストール (Section3.3 参照)」のページからダウンロード出来ます。姉妹 tree の作成に役立ててください。

尚、インストーラを作るには

/Developer/Applications/Utilities/PackageMaker.app
を使います。

6.1 段取り

一般的にインストーラを作成するのに必要な段取りは以下になります。

1. 作業場所を作る。
2. インストールするファイル類を用意する。
3. インストールする手順を所定の各ファイルに記述する。
4. 「PackageMaker」でインストーラを作成する。
5. 「ディスクユーティリティ」でディスクイメージを作成する。

これらの作業は一寸面倒です。
覚え書き程度に書いていきます。

6.2 作業場所を作る

インストーラを作る作業場には
「インストールするファイル類」と「手順を記したファイル類」を置く場所が必要です。

MacOS X WorkShop では、ディレクトリ「OSX-WS」を作業場のルート・ディレクトリとし、
「インストールするファイル類」は「OSX-WS/ROOT」に、
「手順を記したファイル類」は「OSX-WS/Resources」に
置いています。

以下これらのディレクトリ構造を基にして記述していきます。
適宜読み替えて下さい。

²⁸<http://developer.apple.com/documentation/DeveloperTools/Conceptual/SoftwareDistribution/index.html>

6.3 インストールするファイル類を用意する

MacOS X WorkShop のインストーラがすべき事は、**最低限の apt-rpm 環境をつくる事**です。

インストーラのソース一式²⁹の中の make-tree.sh が、必要なファイルを所定の位置にコピーするスクリプトです。RPMDIR=/Users/tkoba/rpm/RPMS/noarch を適宜書き換えた上で、「OSX-WS/ROOT」の中で実行して下さい。

このスクリプトの中でしている事は

1. ディレクトリの作成
2. apt, rpm バイナリー類のコピー
3. 基本パッケージ OSX-{Preferences,X11,system}* のコピー

です。

ただし、基本パッケージのコピーの後、

同一パッケージの複数のバージョンが入っていない事を確認してください。

これで、「OWX-WS/ROOT」以下にインストールされるファイル類が準備されます。

注意！

ファイルのコピー先に**直接 /etc を指定しない**で下さい。

/etc は /private/etc へのシンボリックリンクになっています。

/etc を指定すると PackageMaker の設定によっては/etc のシンボリックリンクを消去して

新たに /etc ディレクトリを作りファイルをインストールしてしまいます。

即ち、**最悪 MacOS X が起動しなくなります**。

6.4 インストールする手順を所定の各ファイルに記述する

ソフトをインストールするには、インストールしようとしている環境が適切であるか確認する必要がありますし、

ファイルを所望の位置に置いた後に、某かのお決まりの設定をする必要がある事もままあります。

ここではその様な手順を実現する方法を述べます。

PackageMaker Help に記述がありますが、インストーラが行う手順は以下になります。

1. InstallationCheck
2. VolumeCheck
3. preflight

²⁹OSX-WS.tar.bz2

4. preinstall or preupgrade
5. (INSTALLER EXTRACTS AND INSTALLS THE PACKAGE'S CONTENTES.)
6. postinstall or postupgrade
7. postflight

MacOS X WorkShop ではこの中の、
InstallationCheck, postinstall, postupgrade を利用
しています。
以下順次説明していきます。

InstallationCheck 「OSX-WS/Resources/InstallationCheck」がスクリプトの実態で、
「OSX-WS/Resources/Japanese.lproj/InstallationCheck.strings」がメッセージ(日本語環境 UTF-8)
の実態です。

ここでは、インストールしようとしている環境が適切であるかを確認します。
している事は、

- MacOS X のバージョンは適切か？
- X11 がインストールされているか？
- Xcode がインストールされているか？

のチェックと、状況に応じたメッセージの表示です。

詳細は **Apple** のドキュメント³⁰ をご覧下さい。

postinstall, postupgrade 「OSX-WS/Resources/postinstall」がスクリプトの実態で、
「OSX-WS/Resources/postupgrade」は、現在 postinstall へのシンボリックリンクです。
Tiger 版のインストーラでは別のファイルに成る予定です。

ここでしている事は、

- rpm database の初期化
- 基本パッケージ OSX-{Preferences,X11,system}* のインストール
- ドットファイルを各ユーザーへ配布

です。

最後に、「OSX-WS/{Welcome,ReadMe,License}.rtf」を作っておきます。

³⁰<http://developer.apple.com/documentation/DeveloperTools/Conceptual/SoftwareDistribution/index.html>

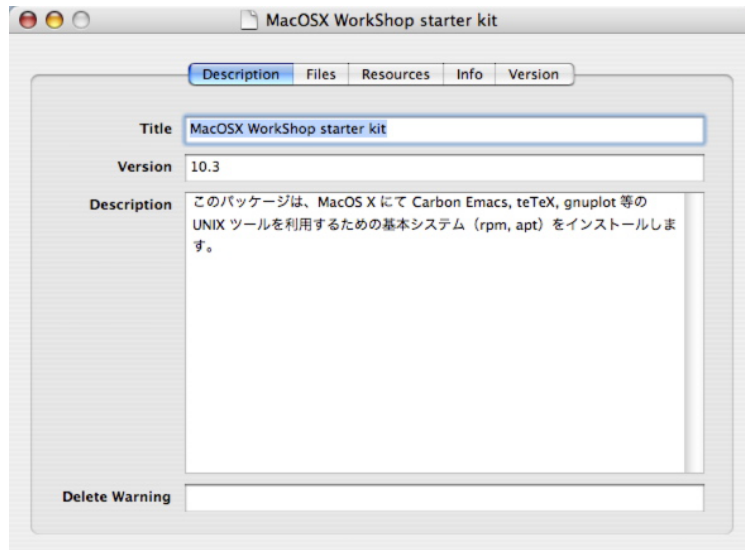


図 1: 「PackageMaker-Description」

6.5 インストーラを作成する

ファイル類の準備ができれば、PackageMaker を使ってインストーラを作ります。

● PackageMaker を起動すると、「Description」タグが選択されたウィンドが現れます。「Title」「Version」「Description」に記述します。

● 「Files」タグではインストールするファイルを置いてあるディレクトリ「OSX-WS/ROOT」を指定し、全体を圧縮する為に「Compress Archive」にチェックを付けます。

● 「Resource」タグではスクリプト・ファイルを置いてあるディレクトリ「OSX-WS/Resources」を指定します。

● 「Info」タグでは以下の設定をしています。

- **Default Location**

インストール先を指定します。

ここではルートディレクトリ「/」を指定します。

- **Post-Install Action**

インストール後の動作を指定します。

login のし直しで大丈夫な筈ですが、大事を取ってここではリスタートするように指定しています。

- **Authorization Action**

インストール時に必要な権限の指定

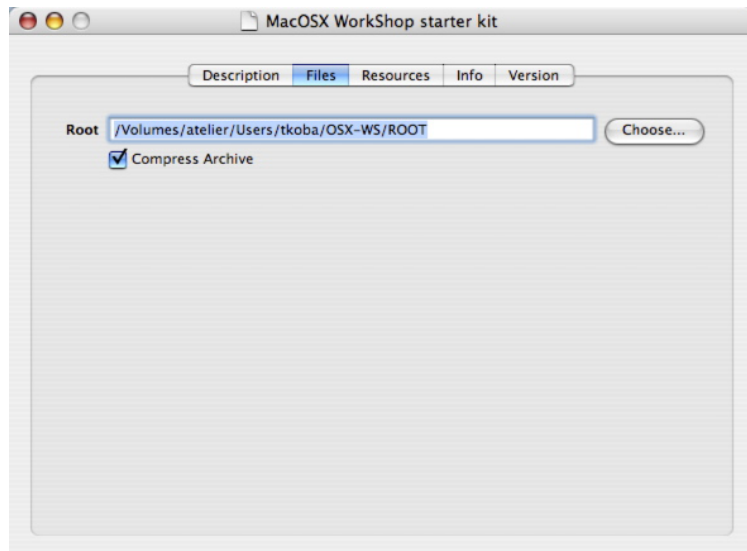


図 2: 「PackageMaker-Files」

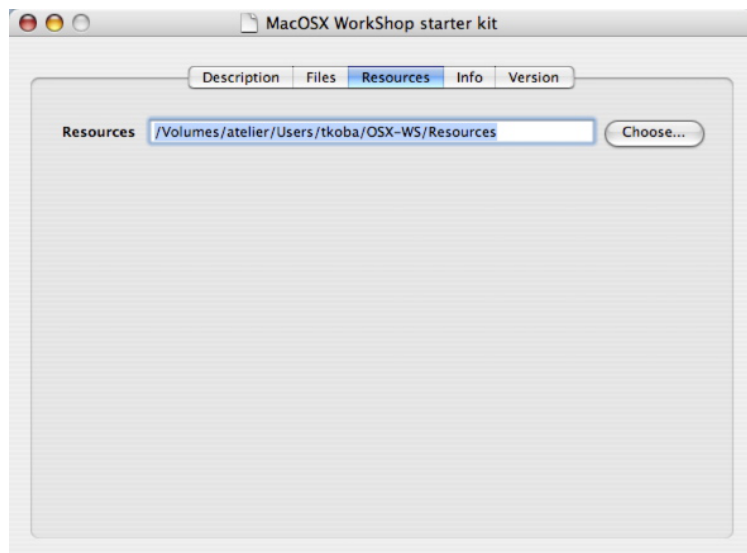


図 3: 「PackageMaker-Resources」

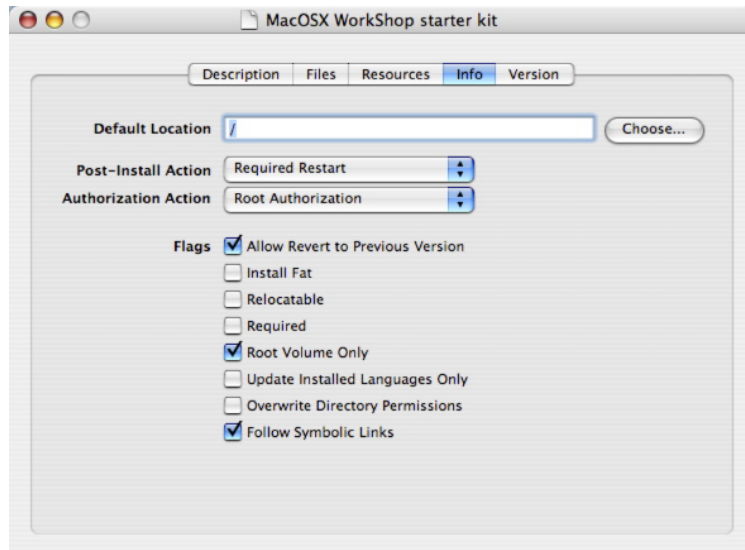


図 4: 「PackageMaker-Info」

root の権限を与えます。

- **Flags: Allow Revert to Previous Version**
インストールしようとしているものよりも新しいものが既にインストールされている場合にインストールを続行させるか否かの設定です。ここでは一応許可しています。
- **Flags: Root Volume Only**
インストール先のボリュームを起動ボリュームに限るか否かの設定です。
当然チェックを付けておきます。
- **Flags: Follow Symbolic Links**
インストール先がシンボリックリンクであった場合に、そのシンボリックリンクを認めるか否かの設定です。
チェックを付けないとそのシンボリックリンクを消してしまうので、チェックを付けておきます。
ただし、これはあくまで保険として付けておくもので、インストール・ファイルを置いておく「OSX-WS/ROOT」以下の段階で
/etc 等シンボリックリンクに向けたディレクトリを作らない事が大事です。

その他のフラグは必要に応じてチェックして下さい。

- 「Version」タグでは、その名の通り version を記述します。
- 全部設定し終わったら保存した後に、「File」→「Create Package」でインストーラを作成します。

6.6 ディスクイメージを作成する

これまでの作業でパッケージのインストーラ MacOSX WorkShop starter kit.pkg が出来ました。

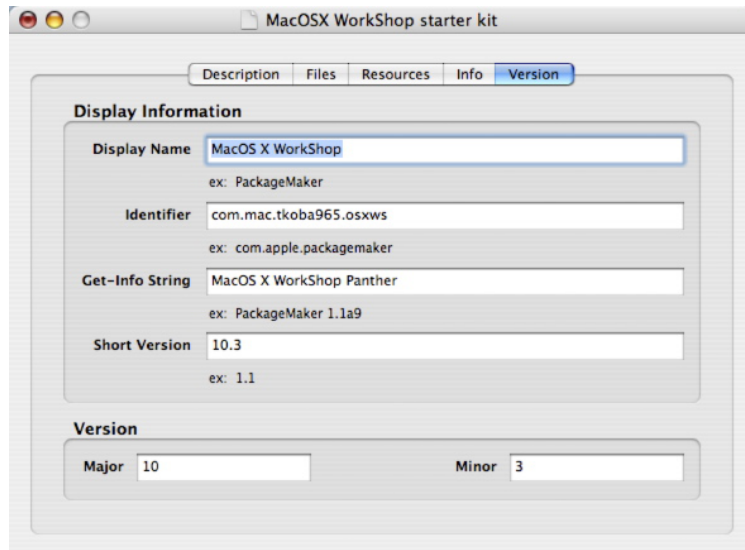


図 5: 「PackageMaker-Version」

後はこれをディスクイメージの中に置いておしまいです。
以下の作業をします。

イメージの作成 「ディスクユーティリティ」を立ち上げて「新規イメージ」ボタンを押し、
ディスクの容量を必要なだけ設定して（私は 7MB にしました）空のイメージを作ります。
空のイメージをマウントして、
そこに ReadMe.rtf と作成したインストーラを入れてマウント解除します。

イメージの圧縮 「ディスクユーティリティ」のメニューから「イメージ」→「変換...」を選択し、
上で作成したイメージを選択します。
「イメージフォーマット」に「圧縮」を選んで保存します。

尚、これらの処理をスクリプトにしてあります。
インストーラのソース `MacOSX-WS-10.3j.tar.bz2`³¹ 中にある `make-pkg.sh` をご覧ください。

³¹MacOSX-WS-10.3j.tar.bz2

7 apt-rpm tree を作る

MacOS X WorkShop の tree を小林が用意した際の備忘録です。
訛度間違いがあるとおもいます。

間違いを発見されましたらご連絡ください。

また、貴方独自の add-on tree を用意される場合もほぼ同じ手順で可能です。

7.1 段取り

一般に apt-rpm tree を作成するのに必要な段取りは以下になります。

1. tree を置く場所を用意する。
2. パッケージを置く。
3. データベースを作る。
4. apt-line を記述する。

これらの作業は web や anonymous ftp に物を置いた経験があれば簡単です。
覚え書き程度に書いていきます。

7.2 tree を置く場所を用意する

tree は「web」と「ftp」と「local disk」に置けます。

ftp も web も local disk も単純にディレクトリを作るだけなので、
ここでは web に置く方法を述べます。

以下の条件を満たしていれば大丈夫です。

- 数百 MB のファイルを置く容量を確保出来るか？
- サーバの性能は十分か？
G4 1GHz 程度の性能があれば、実質問題はないでしょう。

これ以降では tree のルートディレクトリを「OSXWS」とします。

7.3 パッケージを置く

MacOS X WorkShop は現時点では Panther 向けであり、
Tiger がリリースされたら Tiger 用に作り直したパッケージを別途置かなければ成りません。

ですから、先を見越して現在の tree は「OSXWS/Panther」の中に置く事にします。

• ソース・パッケージ

ソース・パッケージ (*.src.rpm) は「OSXWS/Panther/SRPMS.main」に置きます。
このサフィックス「main」はパッケージをカテゴリ分けする際に意味をなします。
即ち、core, devel, plus 等に分類したい場合はそれぞれ「SRPMS.core」などとすれば良い訳です。
MacOS X WorkShop では小さな tree ですから main 一つで十分だと判断し、
カテゴリ分けはしていません。

• バイナリー・パッケージ

バイナリー・パッケージ (*.ppc, noarch).rpm) は「OSXWS/Panther/ppc/RPMS.main」に置きます。

7.4 データベースを作る

tree の実態が置かれたら、データベースを作成します。

MacOS X WorkShop の場合は

```
genbasedir --progress --bz2only [OSXWS]/Panther/ppc main
```

としています。

当然 [OSXWS] は適切なディレクトリに置き換えてください。

この操作はパッケージを更新する度に必要になりますから、
シェルスクリプトにでもしておくとい良いでしょう。

7.5 apt-line を記述する

最後に apt-line を記述する為に apt パッケージを変更します。

```
$ cd ~/rpm/SRPMS  
$ apt-get source apt
```

して、apt のソースパッケージを展開します。

次に、~/rpm/SOPURCES/sources.list-0.5.15-osxws に貴方が作った apt-line を記述します。

既にある記述を真似れば難しくないと思います。

先に /private/etc/apt/sources.list を編集して動作を確認しておき、

正しく動作するものを ~/rpm/SOPURCES/sources.list-0.5.15-osxws にコピーするのが良いでしょう。

最後にスペックファイルを編集（リリース番号の更新と変更履歴を記述）したら、

```
$ cd ~/rpm/SPECS  
$ rpmbuild -ba apt-0.5-tk.spec
```

で所望の新しい apt パッケージが ~/rpm/RPMS/ppc 以下に作られます。

この新しい apt パッケージも忘れずに貴方の tree に加えてください。

8 パッケージメモ

ここでは主に各パッケージに施した変更や拡張について記します。

全てのパッケージについて記述している訳ではありません。

パッケージの詳細は **Panther の apt-rpm tree**³² から全てのパッケージにアクセス出来る他、rpm2html を用いて **RPM 解説データベース**³³ も用意しました。

8.1 Emacs 関連

MacOS X WorkShop では、今のところ CarbonEmacs のみを提供しています。

CarbonEmacs に関する情報は **MacEmacs**³⁴ や、銭谷さんの **Carbon Emacs パッケージ**³⁵ をご覧ください。

収録しているパッケージ内容は銭谷さんのものを基本としています。

ただし、CarbonEmacs 本体と、Emacs Lisp は別のパッケージにして、

独立して更新やメンテナンスが出来る様に配慮してありまし、

elisp の置き場所もパッケージ内部ではなく /usr/local/share/emacs/ 以下に変更しています。

自分独自の elisp (但しバイトコンパイルしていないもの) を置く場合には、
/usr/local/share/emacs/site-lisp/ 以下に置いて下さい。

また、Vine Linux から alternatives を移植してありますから、
XEmacs など他の Emacsen を共存して入れる事も可能 (な筈) です。

関連するパッケージは以下になります。

Mule-UCS MULE-UCS is a coding system and character code translator system.

apel Emacs 用の 基礎的な関数を提供するライブラリ

emacs GNU Emacs エディタ (Carbon 版)

emacs-lisps Carbon Emacs 用の便利な Lisp ライブラリ集

銭谷さんのパッケージに入っている Lisp を纏めたものです。

emacsen-common Common facilities for all emacsen.

³²<http://www.bach-phys.ritsumei.ac.jp/OSXWS/Panther/>

³³<http://www.bach-phys.ritsumei.ac.jp/OSXWS/Panther/rpm2html/>

³⁴<http://macemacs.jp.sourceforge.jp/>

³⁵<http://home.att.ne.jp/alpha/z123/emacs-mac-j.html>

flim Emacsen 用の message に関する表現形式や符号化のためのライブラリです。

mew Emacs でメールを読むためのインターフェース

semi Emacsen 用の MIME の機能を提供するライブラリ

ispell emacs 上で利用できるスペルチェッカー

「Tools」メニューからスペルチェックを選べば利用出来ます。
コンソールからの利用も可能です。

task-emacs emacs バーチャルパッケージ

このパッケージを apt でインストールすると、次のパッケージが自動でインストールされます。
alternatives emacs emacsen-common emacs-lisps apel flim semi Mule-UCS

yatex 野鳥 (YaTeX) - Yet Another TeX mode for Emacs

8.2 T_EX 関連

パッケージの内容は、土村さんと小林が主にメンテナンスしている Vine Linux の teT_EXpackage 群と基本的に同一です。

teT_EX 本体 S-JIS で作成しています。

EUC で利用する為に /usr/local/bin/{platex-euc,ptex-euc} を同梱しています。

VFlib を必要としない様にしてあります。

また、齋藤修三郎さんの OTF パッケージを利用する為に updmap に変更 (tetex-2.0-updmap-otf.patch) を加えて

udvips 用の map file から dvipdfmx 用の map file を自動生成する様にしてあります。

previewer 内山孝憲さんが作成されている Cocoa の dvi previewer です。

MacOS X WorkShop では xdvi ではなく、この Mxdvi をデフォルトにしています。

apt-get でインストールすると、必要なフォントのパッケージ Mxdvi-fonts も同時にインストールされます。

T_EXmacros tetex-macros tetex で用いる追加マクロパッケージ集です。

次のマクロを取録しています。 jsclasses, prosper, epsbox.sty, eclepsf.sty

texmacro-otf 齋藤修三郎さんによる「OpenType Font 用 macro と VF」です。

齋藤氏が配布されているマクロや VF 以外に次の補助ツール類を同梱しています。

updmap-otf

dvipdfmx, udvips 等で埋め込むフォントを設定する為のツールです。

sudo updmap-otf auto とすると

OTF-Hiragino パッケージがインストールされていればヒラギノを埋め込む様に設定し、無ければ noFont の設定をします。

`sudo apt-get install task-tetex` としていれば、デフォルトでヒラギノを埋め込みます。他にも、モリサワ基本7書体パッケージ (OTF-Morisawa-basic7) がインストールされていれば、`sudo updmap-otf morisawa` とすると利用可能になります。

利用方法は `updmap-otf` で表示されます。

`kozuka-udvips.map`

購入したパッケージの小塚ファミリー日本語 OpenType フォントの map file です。

現在は利用出来ません。

`morisawa-udvips.map`

モリサワ基本7書体の map file です。

`noFont-udvips.map`

OpenType Font が全くインストールされていない時に使用される map file です。

フォントを埋め込みたくない時には `sudo updmap-otf nofont` とすれば

このファイルが使われて埋め込まれなくなります。

font 関連 jvf

makejvf

OTF-Hiragino MacOS X 付属のヒラギノフォントを利用する為の設定パッケージです。

OTF-Morisawa-basic7 購入して MacOS X にインストールされたモリサワ基本7書体 OpenType Fonts を利用する為の設定パッケージです。

OTF-Morisawa-RmSgSmg 購入して MacOS X にインストールされた以下のモリサワ OpenType Fonts

A-OTF-RyuminPro-{Regular,Heavy}.otf,

A-OTF-ShinGoPro-{Regular,Heavy}.otf,

A-OTF-ShinMGoPro-{Regular,Bold}.otf

を利用する為の設定パッケージです。

freetype TrueType や OpenType を X11 上で扱う為のエンジンです。

urw-fonts free で品位の高い 35 の標準 PostScript Fonts です。

ttfonts-ja free の日本語 TrueType Font である「さざなみフォント」³⁶ をインストールします。

Mxdvi-fonts Mac 様に内山さんが作られた TeXFonts で、次のフォントがインストールされます。
ComputerModern, amsp-oztex, pxfonts, txfonts

その他 EqEditor \LaTeX Equation Editor³⁷

Apple の Keynote 等で数式を扱う際に利用出来ます。

task-tetex TeX 関連パッケージを簡単にインストールするための仮想パッケージです。

dvipdfmx dvi file を PDF file に変換するツールです。

齋藤さんの OTF パッケージに対応しています。

latex2html TeXfile を html file に変換するツールです。

このドキュメントも latex2html を用いて書かれています。

³⁶<http://sourceforge.jp/projects/efont/files/>

³⁷<http://evolve.lse.ac.uk/software/EquationEditor/>

yatex

8.3 X11 関連

Plotmtv X11 上で動作するグラフ作成ツール

Xaw3d A version of the MIT Athena widget set for X.

freetype A free and portable {True,Open}Type font rendering engine.

ghostscript A PostScript(TM) interpreter and renderer.

VFlib は使用しません。

gnuplot A program for plotting mathematical expressions and data.

VineLinux のものと同様に gnuplot+ の拡張を取り入れています。

gv A X front-end for the Ghostscript PostScript(TM) interpreter.

kinput2 kinput2 - kanji input server for X11

ことえりパッチ適用。

kterm 日本語の表示が可能な X 上のターミナルソフト

openMotif The Open Motif runtime components.

ttfonts-ja Free Japanese TrueType fonts

内容はさざなみフォント³⁸です。

urw-fonts Free versions of the 35 standard PostScript fonts.

xgraph11 xgraph - 2D data plotting program (+ hack 9 + color PS + and so on.)

yaplot yaplot - an easy 3D modeller and animator

8.4 開発関連

rpm The RPM package management system.

詳細は spec file を参照してください。

apt RPM を扱える Debian のパッケージツール apt(Advanced Packaging Tool)

static build して strip してあります。

³⁸<http://sourceforge.jp/projects/efont/files/>

gcc-g77 g77 A GNU Fortran compiler.

HPC MacOS X³⁹ のものです。

8.5 System 関連

OSX-system MacOS X の標準ライブラリやツールを rpm system に教える為のパッケージです。

/private/etc/{profile,csh.login,zprofile} を書き換えて /usr/local/bin, /usr/X11R6/bin 等にパスを通します。

MacOS X WorkShop インストーラによりインストールされます。

OSX-X11 X11 のライブラリやツールを rpm system に教える為のパッケージです。

MacOS X WorkShop インストーラによりインストールされます。

OSX-Preferences OSX-Preferences パッケージは MacOS X WorkShop の基本システムの一部で、デフォルトのユーザ設定ファイル (.Xresources, .bash_logout, .bash_profile, .bashrc) 等を収録しています。

各ユーザーに配布された設定ファイルを更新する為に、osxws-upgrade スクリプトを収録しています。

OSX-Preferences package の更新に対応する為に個人用の記述は **.bashmyrc, .cshmyrc, .zshmyrc, .emacs.my.el** を作成し、**その中に記述して下さい。**

/System/Library/User\ Template/Japanese.lproj/ 内にインストールされますから、新規ユーザー作成時に自動で設定ファイル類がコピーされます。

MacOS X WorkShop インストーラによりインストールされます。

OSX-base rpm system を利用する為に最低限必要なパッケージをインストールする為の仮想パッケージです。

MacOS X WorkShop インストーラを実行した直後に `sudo apt-get install OSX-base` しておく必要があります。

³⁹<http://hpc.sourceforge.net/>

9 スクリーンショット

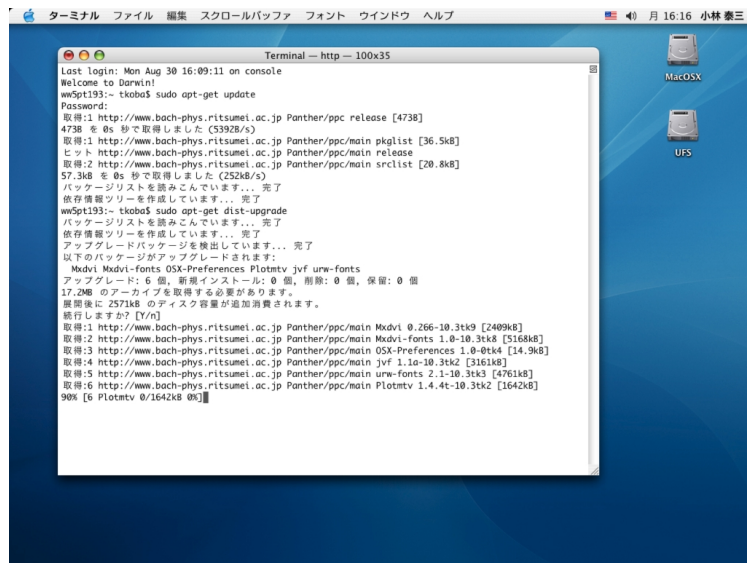


図 6: apt-get で update packages を download 中。

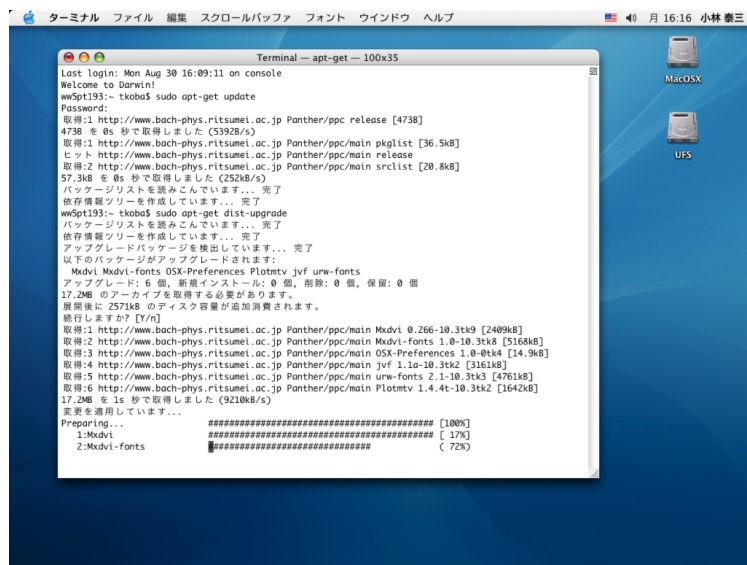


図 7: apt-get でパッケージを更新中。

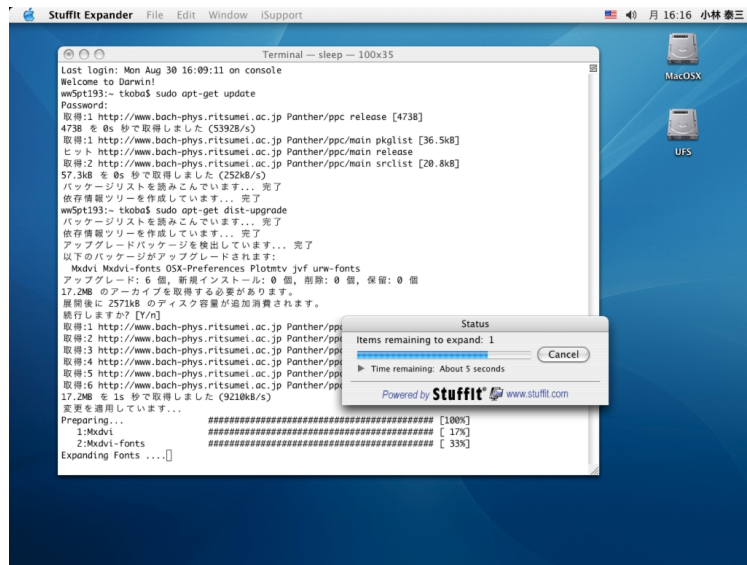


図 8: apt-get でパッケージの更新中に StuffIt Expander が呼び出されているところ。

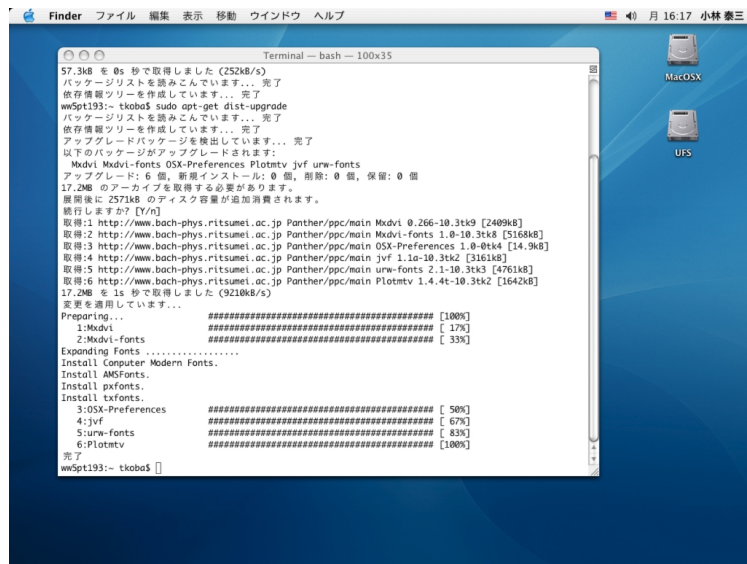


図 9: アップデート終了。

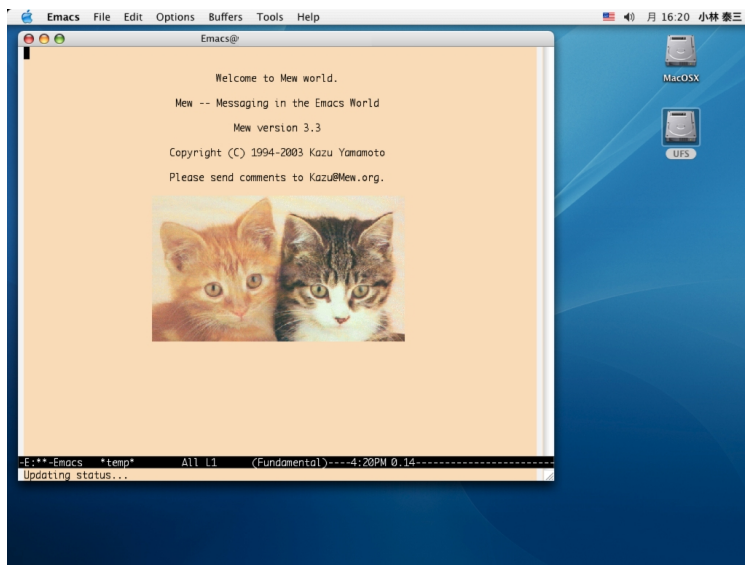


図 10: CarbonEmacs で mew を立ち上げているところ。

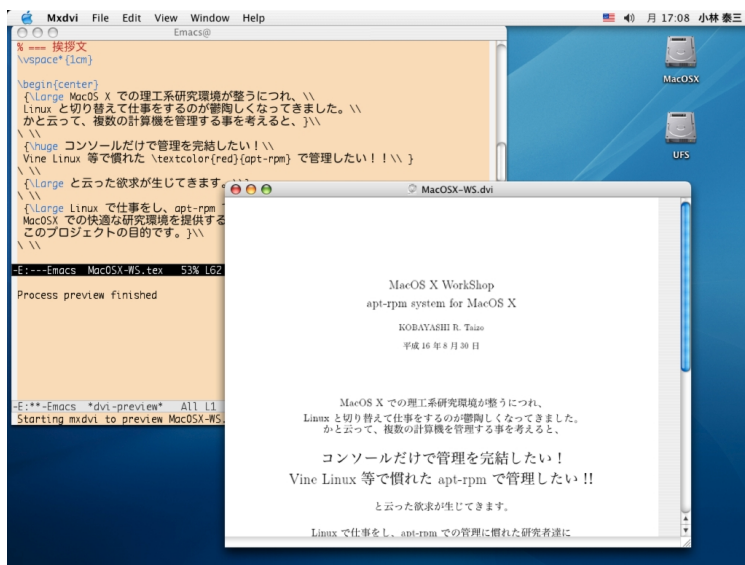


図 11: CarbonEmacs で yateX を用いて LaTeX の文章を書き Mxdvi でプレビューしているところ。Mxdvi からそのまま印刷可能です。

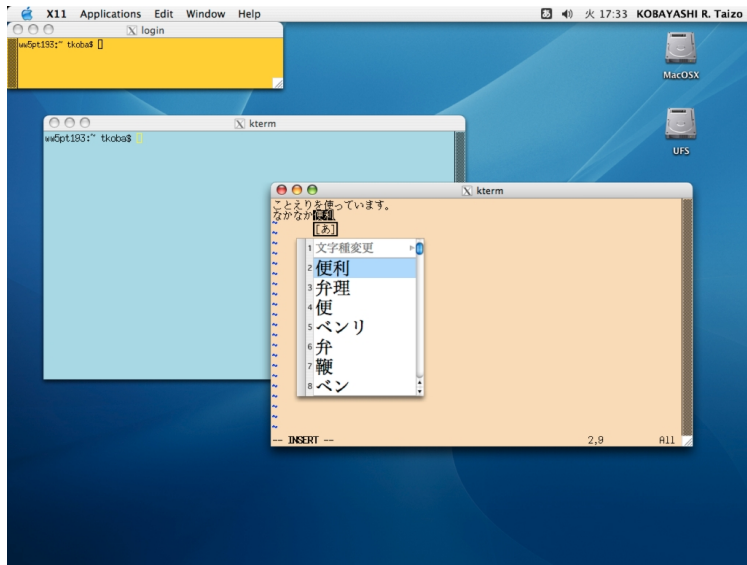


図 12: kterm 上の vim で kinput2 を通して日本語の文章を書いているところ。ことえりパッチを適用してあります。

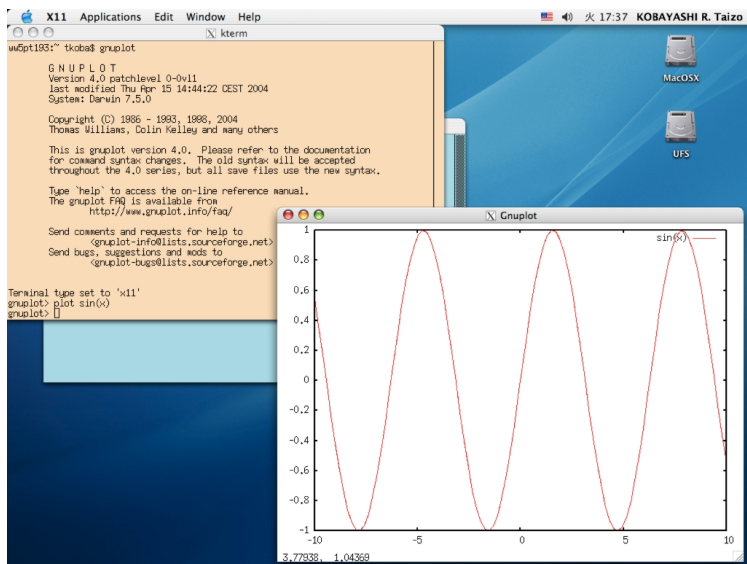


図 13: Vine Linux と同じ gnuplot+ パッチを適用してあります。

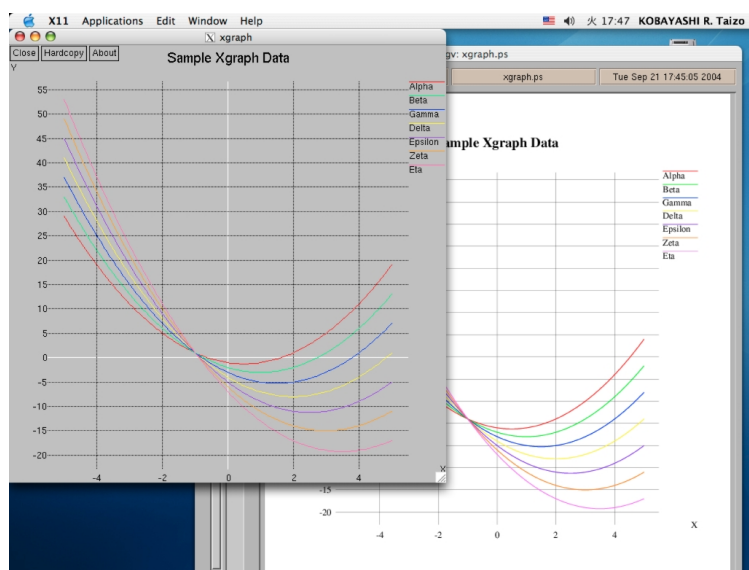


図 14: Vine Linux と同じパッチを適用してあります。Color PS を生成できます。PS ファイルをダブルクリックすれば大抵の場合 PDF に変換でき、プレビューで確認と印刷ができます。

10 既知の問題点と注意点

2005年1月17日現在、MacOS X WorkShop には以下の様な問題があることが分かっています。

emacs と「ことえり」1 CarbonEmacs で「ことえり」を用いた変換中に Auto-Saving が掛かると表示が固まる。

マウスをクリックすると抜けられます。

emacs と「ことえり」2 CarbonEmacs を no window mode (-nw) で起動するとことえりが使えない。

SKK を利用してください。

StuffIt Expander Mxdvi-fonts など一部のパッケージが、インストール時に StuffIt Expander を利用します。

現在 MacOS X に標準で付いてくる 7.0.3 と free の 8.0.2, 9.0 で動作確認しています。

ただし、free のものでも製品版の StuffIt Deluxe 等も

必ず StuffIt Expander を /Applications (アプリケーションのフォルダ) 内に置く
様にインストールしてください。

etc. その他気付いていない問題点があるかもしれません。

11 過去の議論

ここは Mac Wiki⁴⁰ にて行われた過去の議論の書庫です。

11.1 10.3 ~ 10.3-6 公開迄

- iMac G5 にインストールして見ましたが、yatex 利用時に \$HOME がない ex file をコンパイルしようとすると、\$HOME 直下を見に行ってしまう、I can't find file *.tex などとおこられてしまいます。いろいろ調べた結果、.bashrc の KADOBEGYAHOME の設定を外すと問題なくなりました。KADOBEGYAHOME は何のためにあるのですか。Google で検索しても出てこないのわかりません。
 - ご報告有り難うございます。OSX-Preferences-10.3tk9 にて対応しました。
.bashrc 等の dot files は弄られていると osxws-upgrade では自動更新されません。パッケージの更新後に以下の様に手でファイルの更新をして下さい。

```
$ cp /System/Library/User\ Template/Japanese.lproj/.bashrc ~
```

この問題は「emacs のアイコンをクリックして起動した場合に限り」起こる事を確認しました。私は何時も terminal から起動しているので気づきませんでした。環境変数 KADOBEGYAHOME はホームを /Users 以下以外に設定した場合に [kx]term のプロンプトがフルパス表示なるのを避ける為に、極めてアドホックに追加した物です。この設定部分を TERM 変数を読んで振り分ける様にしました。 – tkoba
- w3m や nkf、lv といった小ツール類もインストールできないでしょうか？ 中期的には、rpm 作成という形で貢献できればよいのですが...
 - 意図を書いて下さり有り難うございます。
rpm パッケージの作成や追加は大歓迎です。姉妹 apt-rpm tree を作って下されば更に嬉しいのです。基本的な考え方が異なる tree が増えた方が、より多くの利用者にも開発者にも利益になると思います。さて、w3m ですが、Vine の物を基にして w3m, w3m-el を加えました。nkf は既に入れてあります。lv は加えるよりも less を tune する方針にします。 – tkoba
- OTF-Hiragino パッケージがインストールされていれば、ヒラギノを埋め込むように設定されるようですが、私の環境では nofont の設定がされるようです。MacOS X 10.3.5 の標準で入っているヒラギノは使えないのでしょうか？
 - ご報告有り難うございます。
texmacro-otf-1.2.4-10.3tk6 にて修正しました。フォントマップの設定に関しては \$ updmap-otf でご覧下さい。 – tkoba
- 以前少し触れましたが、初期設定に一切手を付けないインストールが出来ると嬉しいです。設定関連のパッケージは分離できないもののでしょうか？ (/usr/local のみ改変)

⁴⁰<http://macwiki.sourceforge.jp/cgi-bin/wiki.cgi>

– 考えていません。

OSX-system package の存在意義は、システム全体に共通の設定と各ユーザーが設定すべき事項を分離する事と、Vine Linux 等からの移植を容易にする事の二つがあります。これらのメリットを上回る利益が無い限り変更はしません。ですので、まずは初期設定に一切手を付けないインストールが出来ると何が嬉しいのかを説明して下さい。その内容が尤もであれば検討します。

– tkoba

- 設定ファイル (ドットファイル) の更新では、一括で*.rpmnew が作成されるようになっていますが、更新されたファイルだけ作成するようにはできないでしょうか。

– いずれこの要望は上がると思っていました :-P

Vine の vupgrade を手本に osxws-upgrade script として実装しました。OSX-Preferences-10.3-1tk8 に同梱しています。お手数ですが、*.rpmnew の消去と osxws-upgrade の最初の実行だけは手動でお願いします。.bashrc 等を弄っていなければ、これ以降は自動で更新されます。詳細は「インストール (Section3.3 参照)」をご覧ください。– tkoba

- carbon-font の開発版では、Ayuthaya フォントの有無を判別して適切にフォントセットを定義するようになったそうです。近い将来、Ayuthaya.ttf は必須でなくても良くなるかもしれませんね。– ぜ

– コメント有り難うございます。

少なくとも必須にする必要はなくなったので、インストーラと web の表記を作り替えました。

– tkoba

- 10.3-3 で

```
sudo apt-get install task-tetex
```

を実行しますと、途中で

```
6:urw-fonts          ##### [ 26%]
warning: user tkoba does not exist - using root
warning: group tkoba does not exist - using root
warning: user tkoba does not exist - using root
warning: group tkoba does not exist - using root
warning: user tkoba does not exist - using root
.....
```

となります。インストール自体は最後までいくので問題ないのかもしれませんが念のため。

– ご報告有り難うございます。

インストール自体に問題はありません。urw-fonts の spec file には他のパッケージと同様に %defattr(-,root,wheel) してあるにも拘らず、この urw-fonts だけが正しく設定されていないようです。–tkoba

- ドットファイルの件 (消えてしまったようですが) は自然な指摘だと思います。ひとこと注意を添えておくと親切だと思います。(下記の設定ファイルサンプルを DL しても、空のフォルダしか見当たりません。 懼ファイル名がドット (.) ではじまっているので、Finder から見えないのでしょうか。)

- ご指摘有り難うございます。
「インストール (Section3.3 参照)」のページにて注意書きを加えました。 - tkoba
- 最初に設定ファイル {/private/etc/something,\$HOME/.something} を上書き (・バックアップ) なさるのでしたら、ファイルの内容を予め公開された方が良くと思います。 - ぜ
 - ご指摘有り難うございます。
「インストール (Section3.3 参照)」のページにてアクセス出来る様にしました。 - tkoba

11.2 10.3-6 ~ 10.3-7 公開迄

- 2004-11-23: 済みません。上記のアップグレードをした後、警告: waiting for transaction lock とでるようになって apt や rpm が進まなくなってしまいました。
 - これだけでは状況を判断しかねます。パッケージを web から直接持ってきて rpm を用いてもだめだということでしょうか？ 現在インストールされている rpm の version は何でしょう？ <http://www.bach-phys.ritsumei.ac.jp/OSXWS/Panther/ppc/RPMS.main/> から popt-1.9.1-10.3tk3.ppc.rpm rpm-*4.3.2-10.3tk3.rpm をダウンロードして \$ sudo rpm -Uvh -force -nodeps popt* rpm* する事は可能でしょうか？ もしも rpm コマンドが全く使えない場合は、/usr/local/bin/apt を消して、MacOSX-WS-10.3f インストーラで入れ直してみてください。
 - rpm は rpm -q rpm で rpm-4.3.2-10.3tk3 と出ます。上のように popt と rpm を取って来て試しましたが同じ警告が出ました。インストーラでの入れ直しは、「後処理スクリプト実行中」で止まってしまいます。
また、rpm -q emacs で emacs-21.3.50-10.3tk11.4 と emacs-21.3.50-10.3tk11.3
rpm -q freetype-utils で freetype-utils-2.1.9-10.3tk1 と freetype-utils-2.1.9-10.3tk2
と二重に入ってしまったままです。
 - 再起動しましたところ、rpm が正常に動くようになりました。二重に入ってしまったのも、一方を削除することができました。
 - 良かったです。。ただ今 10.3-7 release に向けて再び全パッケージを構築し直しています。この更新を経れば安定すると思います。 - tkoba
- 2004-11-22 (Mon) 09:23:28 長? : a2ps ですが、a2ps hoge とすればファイル hoge が印刷されると思うのですが、うまくいきません。ためしに、vi test.txt でファイルを作り、a2ps test.txt -o test.ps として ps ファイルにし gv test.ps で見てみたのですが、以下のメッセージが出ました。

```

Error: /rangecheck in --%op_show_continue--
Operand stack:
  (\245\346\241\274\245\266\314\276: \351M-^U\267\350\260\267\345\267M-^] \346M-^E\247)
Execution stack:
  %interp_exit .runexec2 --nostringval-- --nostringval-- --nostringval-- 2 %stopped
Dictionary stack:
  --dict:1059/1123(ro)(G)-- --dict:0/20(G)-- --dict:70/200(L)-- --dict:89/200(L)-- --d
Current allocation mode is local

```


- emacs の起動時ですが、Messages バッファに以下のようなログが残っていました。emacs 自身は一応の動作はするので、問題ないのかもしれませんが気になったので。

```
Loading help-fns...done
Loading mwheel...done
(/Applications/Emacs.app/Contents/MacOS/Emacs)
Loading tool-bar...done
Loading image...done
Loading tooltip...done
Loading advice...done
Loading /private/etc/emacs-21.3.50/site-start.d/20apel-init.el (source)...done
Loading /private/etc/emacs-21.3.50/site-start.d/50flim-init.el (source)...done
Loading /private/etc/emacs-21.3.50/site-start.d/51semi-init.el (source)...
Loading mail-mime-setup...done
gnus-mime-setup is not found.
emh-setup is not found.
Loading /private/etc/emacs-21.3.50/site-start.d/51semi-init.el (source)...done
Loading /private/etc/emacs-21.3.50/site-start.d/99emacs-lisps-init.el (source)...done
Loading encoded-kb...done
```

An error has occurred while loading ‘/Users/hogehoge/.emacs.el’:

```
error: No fonts match ‘-apple-courier new-***-normal--16-*--75-75-m-160-mac-roman’
```

To ensure normal operation, you should investigate and remove the cause of the error in your initialization file. Start Emacs with the ‘--debug-init’ option to view a complete error backtrace.

- 以上 2 点のご報告ありがとうございます。これらのエラーはパッケージのインストールか更新に失敗しているからだと思います。修正版 apt-rpm に更新しました。以下の点順に従ってパッケージを入れ直してみてください。

```
$ sudo apt-get update
$ sudo apt-get clean
$ sudo apt-get dist-upgrade --download-only
$ cd /var/local/cache/apt/archives/
$ sudo rpm -Uvh --force --nodeps apt* readline* popt* rpm*
$ sudo apt-get install task-tetex(tetex を入れていなければここは task-emacs) a2ps
```

- apm、apt を更新し task-tetex および a2ps を入れ直しましたが、どちらも解決しませんでした。(長) 2004-11-22 21:28
- 私の手元の 2 台の環境ではこれらの不具合はでていません。seto さんの環境では如何でしょうか？ また a2ps と emacs 以外では不具合は出ていませんか？ – tkoba

- a2ps と emacs に関する問題は、私の環境でも再現していません。昨日、修正の途中の段階では gnuplot が立ち上がらないということがありましたが、その後更新することで解決しました。あと気がついたことは、おそらく CarbonEmacs のバージョン (2004-11-21 build) の問題でしょうが、Direx モードからファイルを開くことができないというくらいです。- seto
- ご報告ありがとうございます。今日の CVS で試してみます。- tkoba
- 2004-11-20 (Stu) 06:00:00 石? : apt-get update, apt-get upgrade で以下のような error を吐きました。当方の StuffIt? の問題でしょうか。

```

15:Mxdvi-fonts                ##### [ 83%]
Expanding Fonts
/private/var/local/tmp/rpm-tmp.39499: line 33: 24455 Bus error      open -a StuffIt\
Expanding amsp-s-oztex.sitx .....
/private/var/local/tmp/rpm-tmp.39499: line 33: 24503 Bus error      open -a StuffIt\
Expanding computer-modern.sitx .....
/private/var/local/tmp/rpm-tmp.39499: line 33: 24526 Bus error      open -a StuffIt\
Expanding pxfonts-macos.sitx .....
.....

```

- そうですね。ここでは単純にスクリプトで StuffIt Expander を起動しているだけですので、ここで Bus error が起こっているとすると貴方の環境の問題である可能性が高いですね。お使いの StuffIt の version と install path をお教えてください。- tkoba
- Stuffit 8.0.2J を使っていて上の Error が出ました。9.0 や 8.0.2 を何度かインストールして試しているうちに直りました。 .sitx の関連付けがうまくいっていなかったのかもしれませんが。- 石
- 経過のご報告ありがとうございます。その後私の手元でも確認してみたのですが状況の再現はできませんでした。なにはともあれアップデートできて良かったです。今後も何かありましたらご報告くださいませ。- tkoba
- 2004-11-09 (Tue) 21:52:36 ぜ : elisp の内訳はどうなっているのでしょうか？ (たぶん、私の版とは一致しないのではないかと...)
- 勿論完全に一致している訳ではありませんが、Carbon Emacs は銭谷さんのパッケージがデファクトスタンダードであると認識していますので、基本的な使い方をしてる限り同等と考えて差し支えない様にしてある心算です。
銭谷さんのこのご指摘は elisp 云々以前に OSX-WS の内容が、インストールしないとよく判らないのが問題だと解釈しました。そこで rpm2html を用いて全てのパッケージ内容を確認できるようにしました。これによりパッケージの不備が見つかると思います。その際はこれまで同様ご指摘を宜しくお願い致します。- tkoba
- 迅速な対応恐れ入ります。小林さんのシステムは、MacOS の UNIX 部分にソフトをインストールしますから、なるべく中身を透明にした方が良いと思います。- ぜ
- 2004-10-29 (Fri) 18:15:10 石? : apt-get install w3m すると

```
warning: user tkoba does not exist - using root
```

と警告がでます。

- ご報告有り難うございます。どう云う訳か urw-fonts では出ていましたが w3m でも出ましたか。実害は無いのですが鬱陶しいですね。rpm のバグだと思いますが fix する時間は暫く取れそうにないです。(それともヴォリュームを case sensitive にしているのが問題なのかなあ。。)

- tkoba

rpm-4.3.2 で修正されているようです。 - tkoba

- 2004-10-28 (Thu) 04:35:20 む? : bzip2 は不要かもしれません。(Panther に入っている)

- bzip2 パッケージは rpm 自体のパッケージを作る際に /usr/lib/libbz2.a では都合が悪かったからだと記憶しています。Tiger では rpm のバージョンを更新する予定です。その折に不必要であれば bzip2 パッケージは抜こうと思います。 - tkoba

- 2004-10-27 (Wed) 16:22:58 ys? : dvipdfmx でヒラギノの埋め込みがされません。過去の議論で texmacro-otf-1.2.4-10.3tk6 で修正されたとのことですが、その修正版でも埋め込まれません。環境は MacOS X 10.3.5 に task-tetex のインストールです。どのファイルを見たら等というサジェスションはありませんか？

- この情報だけでは状況を再現出来ません。\$ sudo updmap-otf status でどう表示されますか？推測ですが以下の二つのどちらかが問題でしょう。それでも解決しない場合は、

```
$ sudo apt-get remove tetex
```

```
$ sudo apt-get install task-tetex
```

してみてください。 - tkoba

- * TeX file のプリアンブルに sepacageotf が無い。

- * 文字コードが EUC で platex-euc でコンパイルしていない。

- ご対応ありがとうございました。task-tetex の再インストールで無事に埋め込まれるようになりました。再インストールの前後ともに \$ sudo updmap-otf status の結果は以下の様に正常に見えます。手で設定ファイルを変更した記憶はないのですが、どこかを壊していたのかもしれませんが。お騒がせいたしました。(ys)

```
$ sudo updmap-otf status
```

```
Standby map file : hiragino
```

```
Standby map file : noFont
```

```
And INSTALLED map file : hiragino
```

- 経過のご報告有り難うございます。ptex-3.1.4 に更新した際に tetex-2.0.2-10.3tk4 パッケージにバグが入り (Vine のパッケージにも enbug してしまっていました。。) それが修正版の 10.3tk5 の更新時に何かの拍子でうまく除去出来なかったのかもしれない。パッケージ更新時の振る舞いを再チェックしておきます。 - tkoba

- 2004-10-25 (Mon) 08:25:56 ぜ : ライセンスに関する説明 (付属文書 or 解説 WWW ページ) はどこにあるのでしょうか・・・？

- ご指摘ありがとうございます。パッケージに関しては各パッケージに同梱されており、\$ rpm -qi hoge でも参照可能です。

インストーラに関してはインストーラに ReadMe.rtf License.rtf が同梱されています。
ご指摘により、License.rtf を書き換え GPLv2 である事を明示しました。
web page に「1.5 ライセンス」ページを追加しました。 - tkoba

11.3 10.3-7 ~ 10.3-8 公開迄

- 2005-05-01 (Sun) 17:08:25 p03h? : アプリケーション「mdview」についてです。アプリケーションウインドウ上の「Convert」をクリックして開くウインドウ上の、ボタン名がすべて表示されません。ボタン自身は見えているのですが。

- 報告有り難うございます。原因は gtk が locale を正しく扱っていない事にあるようです。取り敢えず以下の様に実行して回避してください。 - tkoba

```
$ LC_ALL=C mdview [options] hogehoge.dat
```

- 上の方法で実行する事で、ちゃんとボタン名が表示されました。ありがとうございます。 -p03h?

- 2005-04-25 (Mon) 20:12:04 fu7mu4? : しばらくすると、Apple から OS 10.4 がでますよね。そのときこの OSXWS は 10.4 に移行するのでしょうか、それとも 10.3 には対応し続けてくれるのでしょうか?

- 10.4 への対応は必ずします（研究上必須である為）が、10.3 とは独立した tree になる為 10.3 はそのまま残ります。10.3 tree を何時迄メンテナンスするかは明言しかねますが、今年度中はアクセス出来る様にする予定です。 - tkoba

- 安心しました。これからも利用させていただきます。 - fu7mu4?

- 2005-04-04 (Mon) 21:29:01 fu7mu4? : apt-get で Emacs を install し直したら、ことえりでの日本語入力ができなくなりました。Error として Can't Activate input method 'Japanese' と出ます。

- 返事が遅くなり失礼致しました。apt-get dist-upgrade されたのであれば OSX-Preferences-10.3-1tk25 も同時に更新された筈です。そして、\$ osxws-upgrade を実行すれば、~/ .emacs.el が更新されて、引き続きことえりを利用できる筈です。もしも、.emacs.el が更新されていなければ、以下を実行してみてください。 - tkoba

```
cp /System/Library/User\ Template/Japanese.lproj/.emacs.el ~
```

- fu7mu4? : \$ osxws-upgrade を実行したのですが、Emacs でことえりは利用できませんでした。上記のファイルをコピーすることで、ことえりによる入力できました。

- 多分 ~/ .emacs.el を編集されていて自動更新がされなかったのだと思います。osxws-upgrade は login 時に自動実行されますから普通は明示的に手で実行する必要はありません。 - tkoba

- どうもありがとうございました。 - fu7mu4?

- 2005-02-25 (Fri) 00:33:12 ぜ : bash_completion をお勧めします。（→ bash をご参照下さい）\$ ssh でタブを押すと .ssh/known_hosts に登録されているホスト名が表示される（推定）など、bash の補完機能が大幅に賢くなります。

- 何時も有益な情報を有り難うございます。これまで tcsh や bash を若干のカスタマイズで利用して特に不便を感じた事がなかったので見過していました。早速試してみて、導入を検討します。- tkoba
- 要望です。Mew 以外に、Wanderlust <http://www.gohome.org/wl/index.ja.html> もパッケージとして追加して頂けませんか？
 - packaging 自体は簡単なのですが、メンテナンスする人が居ないのが問題です。もしも貴方が packaging とメンテを引き受けて下さるのであれば大歓迎ですが、現状 (mew 以外に wl も用意すべき理由や説得を受けていない状況) では消極的です。wl を用意する積極的な理由をお教えください。それが尤もであれば考えます (し、私自身が wl に移行するかもしれません ;-P)
 - tkoba
- 2004-11-24 (Wed) 06:53:47 ぜ : <http://www.apple.com/jp/downloads/macosx/index.html> からメールで申請できるようです。
- 2004-11-24 (Wed) 06:52:48 ぜ : <http://www.apple.com/jp/downloads/macosx/tools/index.html> に掲載を申し込まれてはどうでしょうか？
 - 情報ありがとうございます。10.3-7 release が終わりましたら申し込もうと思います。- tkoba
 - 問い合わせのメールを含めて3回程メールを送っているのですが、一向に音沙汰ないですね。。- tkoba
 - 僕も先月から、Emacs パッケージのタイトル変更を申請しています。今日付けで更新があったようですが、何も変更されていないようです。北米サイトだとその日のうちに変更が反映されるのですが... - ぜ
 - コメント有り難うございます。気長に構える事にしようと思います。OSXWS としては、この MacWiki の御陰で状況把握と整理が必要十分に出来ているのでとても満足しています。- tkoba
 - ようやく Emacs パッケージのタイトルを変更していただきました。現在は、月に1・2回の更新のようです。かなり気長に構えないといけませんね... - ぜ
 - ご連絡有り難うございます。私も早速問い合わせのメールを送っておきました。あまり期待せずに待つ事にします。橋本さんの新しいパッチが今週中に出る様ですから、その後の銭谷さんのパッケージに合わせて 10.3-8 をリリースしようかと考えています。- tkoba

12 謝辞

MacOS X WorkShop は、以下の個人/団体 (順不同) に多大な御指南/御協力を戴いたり、公開されているパッケージや議論を参考にさせて頂きました。

この場を借りて関係各位に感謝の意を表します。

藤井恵介さん	MacOS X WorkShop の下地を築いてくださいました。
内山孝憲さん	Mxdvi とそのフォントパッケージ
齋藤修三郎さん	OTF パッケージ
銭谷誠司さん	CarbonEmacs パッケージ、 Mac Wiki
kenichi kikuchi さん	kinput2 ことえりパッチ
MacWiki	-
Project PINEAPPLE の皆さん	-
Vine Linux の皆さん	-

更新履歴

• Thu Jul 14 2005 KOBAYASHI Taizo

– Version 10.3-8

– Panther 対応版の最終リリース

– パッケージ追加情報

* bash-completion

bash の補完機能を強化するシェル関数集。ざっと見た所かなりの関数が Linux に特化したもの（例えば `insmod` なんて OSX ではまず使わない）のようです。不具合があったり変更した方が都合が良いものがありましたらバグやコメントを宜しくお願いします。

* f2c, pgplot

pgplot は Fortran から利用するグラフィックライブラリです。f2c は Fortran source を C source に変換するツールで pgplot を利用する為に用意しました。gcc-g77 と /usr/bin/ld の相性があまり良くなく、restFP, saveFP が Undefined Symbols になります。f2c 由来の f77 を用い `-lg2c` (gcc-g77 由来) で利用出来ます。

* povray, mdview

mdview は名古屋大学の馬場さんが分子動力学シミュレーション用に開発された 3 D アニメーションビューワーです。

* clisp-2.33.2, maxima-5.9.1, imaxima-0.9

学部学生から *Mathematica* 導入の要望が上がっていましたが、予算が無いので代替候補として Maxima 関連をパッケージングしました。使用に際しては Maxima のページを参照すると共に以下の点をご注意ください。– tkoba

・ imaxima は emacs の fontset-hiraginomaru14 に合わせて (`setq imaxima-scale-factor 1.5`) に設定してあります。

・ emacs をアイコンをクリックして起動した場合、以下の問題があります。

– M-x imaxima すると以下のエラーが出る

```
*** - invalid byte #xE3 in CHARSET:ASCII conversion
```

```
Break 1 [2]>
```

ここで

```
quit
```

とすると抜けられる。

~/gnuplot に以下の設定をしておかないとグラフが描けない。

```
set terminal aqua
```

* PDFlib-Lite-6.0.0

gnuplot で pdf ファイルを扱える様にする為に導入しました。ただ、pdf の書き出しはかなり不安定です。pdflib には libpng, libtiff, libjpeg が同梱されており、システムに既にインストールされているパッケージと干渉する恐れがあるので libpng はシステムのものをリンクする様に作り直しました。libtiff, libjpeg は pdflib 特有の変更がされているのでそのまま利用しました。従って、libpdf を利用するバイナリを作るときは

```
LDFLAGS = -lpdf -lpng
```

が必要です。また libpdf と libtiff, libjpeg を同時にリンクしない様に配慮することが必要です。

– パッケージ更新情報

* emacs-lisps-1.0-10.3tk13

emacs-lisps-init.el の修正

– carbon-emacs-preview -i mac-key-mode への移行を反映 (Thanks 高井さん)

– carbon_font-1.0.2

* OSX-Preferences-10.3-1tk19

– default の emacs color theme を wheat から gnome2 に変更。

– .emacs.el を修正して no window mode での利用を改善。リモートでの `$ emacsnw -e mew` 等を可能にしました。

* a2ps

– default の出力先をプリンターから標準出力に変更しました。

– C 等のソースコードを a2ps → ps2pdf で PDF に変換する wrapper script を a2pdf の名前で用意しました。

```
$ a2pdf --prologue=color hoge.c
```

で hoge.c.pdf が作成されます。

```
$ open hoge.c.pdf
```

して preview か Adobe Reader で印刷して下さい。

• Wed Jan 19 2005 KOBAYASHI Taizo

– 各ページの文章の誤りを訂正。

- Thu Nov 25 2004 KOBAYASHI Taizo

- Version 10.3-7
- Installer ver. 10.3f
rpm-4.3.2 をはじめとする全パッケージの更新。
- **パッケージ追加情報**
 - * Ngraph-6.3.30-10.3tk1
 - * xgraph-12.1-10.3tk1
- xgraph11 から修正パッチを移植しました。百害あって一利無しのアニメーション機能は削除してあります。
- **パッケージ更新情報**
 - * OSX-Preferences-10.3-1tk12
- .bashmyrc, .cshmyrc, .emacs.my.el, .zshmyrc の追加。.*myrc や .emacs.my.el を既にある人は以下のディレクトリから該当するファイルを参照して書き直して下さい。
/System/Library/User Template/Japanese.lproj/
 - * emacs-21.3.50-10.3tk11.5
- CVS 20041123, inline_patch-20041101
 - * OSX-Preferences-10.3-1tk16
- fix osxws-upgrade script
 - * kterm-6.2.0-10.3tk5
- background:wheat, foreground:black に設定
 - * kinput2-v3.1-10.3tk2
- Cmd+Space で日英切り替え出来るように設定。modeLocation を kterm の左下に出るように設定。

- Thu Nov 11 2004 KOBAYASHI Taizo

- rpm2html による RPM データベースのページを追加
- 各ページの文章の誤りを訂正。

- Tue Oct 26 2004 KOBAYASHI Taizo

- Installer の ReadMe.rtf, License.rtf を書き換え GPLv2 であることを明示。
- Subsection 「ライセンス」追加

- Sun Oct 24 2004 KOBAYASHI Taizo

- Version 10.3-6
- Installer ver. 10.3d
gettext, beecrypt, bzip2, OSX-Preferences 更新に伴う更新。
- Section 「過去の議論」を追加。

- **パッケージ追加情報**

- * w3m, w3m-el
kterm 上で画像を表示する場合は w3m-img をインストールして下さい。
- * gtk+, glib, gdk-pixbuf, imlib, libungif
w3m-img の為に導入。
- * OSX-keyring
パッケージに gpg 署名をする為の鍵束。
- * kotonoko
コトノコ⁴¹ ver 1.0-beta26
- * vim
vim-6.3.31 (huge, big, normal)
kterm 上で利用する vim
terminal での日本語入力はダメ。

- **パッケージ更新情報**

- * emacs-21.3-10.3tk10
- CVS 20041024, inline_patch 20041015
- * tetex-2.0.2-10.3tk5
- remove TEXMF/dvips/base/config.ps

⁴¹<http://www.afternooncafe.jp/kotonoko/>

* OSX-Preferences-10.3-1tk10
- added rpm/BUILD dir

• Wed Oct 13 2004 KOBAYASHI Taizo

- Version 10.3-5
- Installer ver. 10.3c
carbon-font.el の改訂に伴い Ayuthaya.ttf に関する記述を変更。
- dot files の更新
OSX-Preferences 更新の際に各ユーザーの設定ファイルを更新する osxws-upgrade script を同梱。

• Tue Oct 12 2004 KOBAYASHI Taizo

- Version 10.3-4
- .emacs.el の更新
font-lock の導入と YaTeX 使用時の skk 環境の整備
- urw-fonts をインストールする際の warning についてを「10 既知の問題点」に追加

• Sun Oct 10 2004 KOBAYASHI Taizo

- Version 10.3-3
- Installer ver. 10.3b
postinstall script で無駄な *.rpmorig を作らない様に修正
- .emacs.el の更新
bibtex-command "jibibtex -kanji=euc" 追加 (坂田君)
"set-terminal-coding-system" を 'utf-8 から 'euc-jp-unix へ変更
terminal や kterm で -nw mode を利用できる様にしてみました。
ただし、ことえりではなく SKK を利用して下さい。
- Mxdvi-fonts の更新
オリジナルの *.hqx を *.sitx で作り直しました。

• Thu Oct 07 2004 KOBAYASHI Taizo

- Version 10.3-2
- Installer ver. 10.3a
- skk, skkdic, skktools 追加
- OSX-Preferences-10.3-1tk5
fixed typo in .bashrc
- emacs-21.3.50-10.3tk7
cvs-20041005
- texmacro-otf
updmap-otf ver. 0.5
利用可能な font map のみを status で表示する様に修正

ToDo

- .emacs.el 関連の更なる調整。
これは Mac Wiki で議論し乍ら発展させよう。

• Wed Sep 29 2004 KOBAYASHI Taizo

- Version 10.3-1
- 設定ファイル {/private/etc/something, \$HOME/.something} の内容を追加。(Thanks. 銭谷さん)

ToDo

- .emacs.el 関連の更なる調整。
これは Mac Wiki で議論し乍ら発展させよう。

• Thu Sep 23 2004 KOBAYASHI Taizo

- Version 10.3
公開版
- apt-rpm tree の作成方法を追加

ToDo

- .emacs.el 関連の更なる調整。
これは Mac Wiki で議論し乍ら発展させよう。

• **Wed Sep 22 2004 KOBAYASHI Taizo**

- Version 1.0
- installer の作成方法を追加

ToDo

- .emacs.el 関連の更なる調整。
これは Mac Wiki で議論し乍ら発展させよう。

• **Mon Sep 20 2004 KOBAYASHI Taizo**

- Version 0.99
- installer の version を 10.3 へ変更。
- zsh の設定ファイルを追加 (新山君)
- pTeX3.1.4, mendex-2.5a, etc..
- emacs Sep 19 CVS

ToDo

- .emacs.el 関連の更なる調整。

• **Tue Sep 07 2004 KOBAYASHI Taizo**

- Version 0.9
- スクリーンショット追加。
- パッケージメモ以外はほぼ完成？

ToDo

- installer の version を 10.3 へ変更。
- .emacs.el 関連の更なる調整。

• **Mon Aug 23 2004 KOBAYASHI Taizo**

- 最初の版

索引

- .rpmmacros, 22
- L^AT_EX Equation Editor, 37

- aclocal, 24
- apel, 35
- apt, 16, 38
- apt-cache, 17
- apt-get
 - clean, 19
 - dist-upgrade, 18
 - install, 17
 - remove, 18
 - upgrade, 18
- apt-get update, 16
- autoconf, 24
- autoheader, 24
- automake, 24
- autotools, 24

- c++, 25

- dvipdfmx, 37

- Emacs, 35
 - CarbonEmacs, 35
- emacs, 35
 - apel, 35
 - emacs-lisps, 35
 - emacsen-common, 35
 - flim, 36
 - ispell, 36
 - mew, 36
 - Mule-UCS, 35
 - semi, 36
 - task-emacs, 36
 - yatex, 36
- emacs-lisps, 35
- emacsen-common, 35
- EqEditor, 37

- flim, 36
- freetype, 37, 38

- g77, 39
- gcc-g77, 39
- ghostscript, 38
- gnuplot, 38
- gv, 38

- Installer
 - InstallationCheck, 28
 - postinstall, 28
 - postupgrade, 28
- ispell, 36

- jvf, 37

- kinput2, 38
- kterm, 38

- LaTeX, 36
- latex2html, 37
- libtool, 24

- macro, 23
- makejvf, 37
- Making Installer
 - Apple's site, 26
- Making RPM
 - Momonga Linux Specfile-Guidance, 22
 - Vine Linux, 22
- mew, 36
- Mule-UCS, 35
- Mxdvi, 36
- Mxdvi-fonts, 37

- openMotif, 38
- OSX-base, 39
- OSX-Preferences, 39
- OSX-system, 39
- OSX-X11, 39
- osxws-upgrade, 46
- OTF-Hiragino, 37
- OTF-Morisawa-basic7, 37
- OTF-Morisawa-RmSgSmg, 37

- pLaTeX, 36
- Plotmtv, 38
- pragma, 25
 - implementation, 25
 - interface, 25
- pTeX, 36
- rpm, 19, 38
 - e, 21
 - i, 21
 - ivh, 21
 - q, 19
 - qa, 19
 - qi, 19
 - qp, 19
 - U, 21
 - Uvh, 21
- semi, 36
- tag, 23
- task-emacs, 36
- task-tetex, 37
- teTeX, 36
- tetex, 36
- tetex-macros, 36
- TeX, 36
 - dvipdfmx, 37
 - jvf, 37
 - latex2html, 37
 - makejvf, 37
 - Mxdvi, 36
 - Mxdvi-fonts, 37
 - OTF-Hiragino, 37
 - OTF-Morisawa-basic7, 37
 - OTF-Morisawa-RmSgSmg, 37
 - task-tetex, 37
 - tetex, 36
 - tetex-macros, 36
 - texmacro-otf, 36
 - yatex, 38
- texmacro-otf, 36
- ttfonts-ja, 37, 38
- urw-fonts, 37, 38
- Vine Linux, 5
- warning
 - user tkoba does not exist, 50
- X11, 38
- Xaw3d, 38
- xgraph11, 38
- yaplot, 38
- yatex, 36, 38
- ライブラリ, 24
 - libiconv, 24
 - libintl, 24