

Migrating from Linux to Mac OS X

David Wheeler
Kineticode, Inc.

Why Mac OS X

- It's Unix
- It Just Works
- It runs common productivity applications
- It Rocks
- It's a joy to Work with
- It's Unix
- Did I mention that it *Rocks*?

Mac User

- Mail
- Office
- Chimera
- iCalendar
- Finder



Power User

- Developer Tools
- Project Builder
- Apache
- CVS
- tcsh



Open Source
means anyone
can join the party
as an equal participant

Alpha Geek

- Perl, Java, Python, Ruby, C, Objective C
- Carbonized Emacs/Vim
- CVS over SSH
- Apache/mod_perl/mod_ssl
- PostgreSQL/MySQL
- Xwindows
- Sendmail



Carbonized Emacs

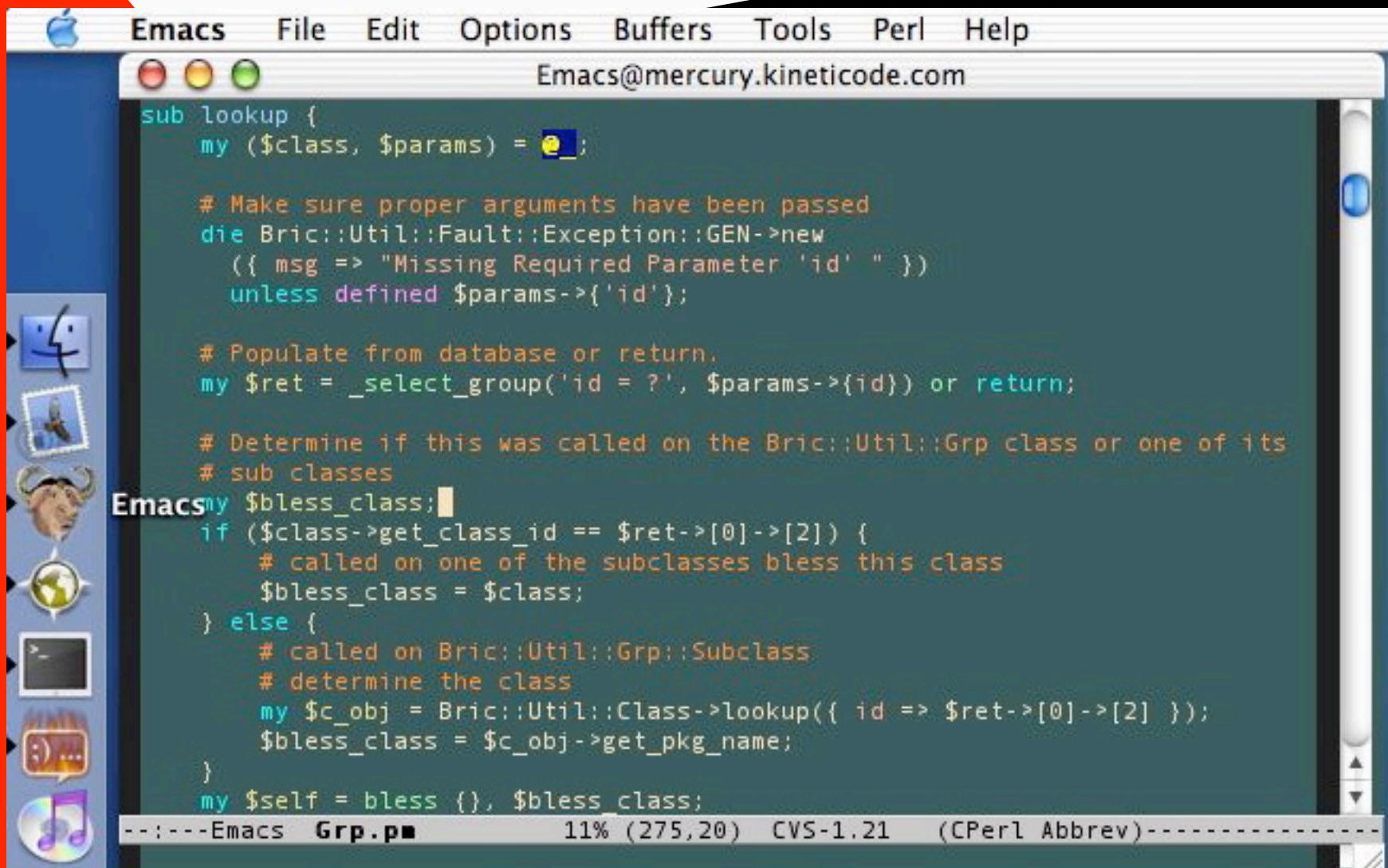
- Ported to Carbon by Andrew Choi
- Available in GNU CVS
- Binaries Available
- Download and build like this:



Carbonized Emacs

```
% cvs - \  
  d:pserver:anoncvs@subversions.gnu.org:/cvsroot/emacs \  
    login  
% cvs -z3 - \  
  d:pserver:anoncvs@subversions.gnu.org:/cvsroot/emacs \  
    co emacs  
% cd emacs  
% ./configure  
% make bootstrap  
% sudo make install  
% sudo cp -rf mac/Emacs.app /Applications
```

Carbonized Emacs



The screenshot shows an OS X desktop environment with an Emacs window open. The window title is "Emacs@mercury.kineticode.com". The menu bar includes "Emacs", "File", "Edit", "Options", "Buffers", "Tools", "Perl", and "Help". The toolbar has three icons: a red circle, a yellow square, and a green triangle. The status bar at the bottom shows "Emacs Grp.pm" and "11% (275,20) CVS-1.21 (CPerl Abbrev)". The left sidebar contains icons for a file, a person, a globe, a terminal, a search function, and a musical note.

```
sub lookup {
    my ($class, $params) = @_;

    # Make sure proper arguments have been passed
    die Bric::Util::Fault::Exception::GEN->new
        ( { msg => "Missing Required Parameter 'id' " } )
        unless defined $params->{ 'id' };

    # Populate from database or return.
    my $ret = _select_group('id = ?', $params->{id}) or return;

    # Determine if this was called on the Bric::Util::Grp class or one of its
    # sub classes
my $bless_class;
    if ($class->get_class_id == $ret->[0]->[2]) {
        # called on one of the subclasses bless this class
        $bless_class = $class;
    } else {
        # called on Bric::Util::Grp::Subclass
        # determine the class
        my $c_obj = Bric::Util::Class->lookup( { id => $ret->[0]->[2] } );
        $bless_class = $c_obj->get_pkg_name;
    }
    my $self = bless {}, $bless_class;
```

Carbonized Emacs Home Page

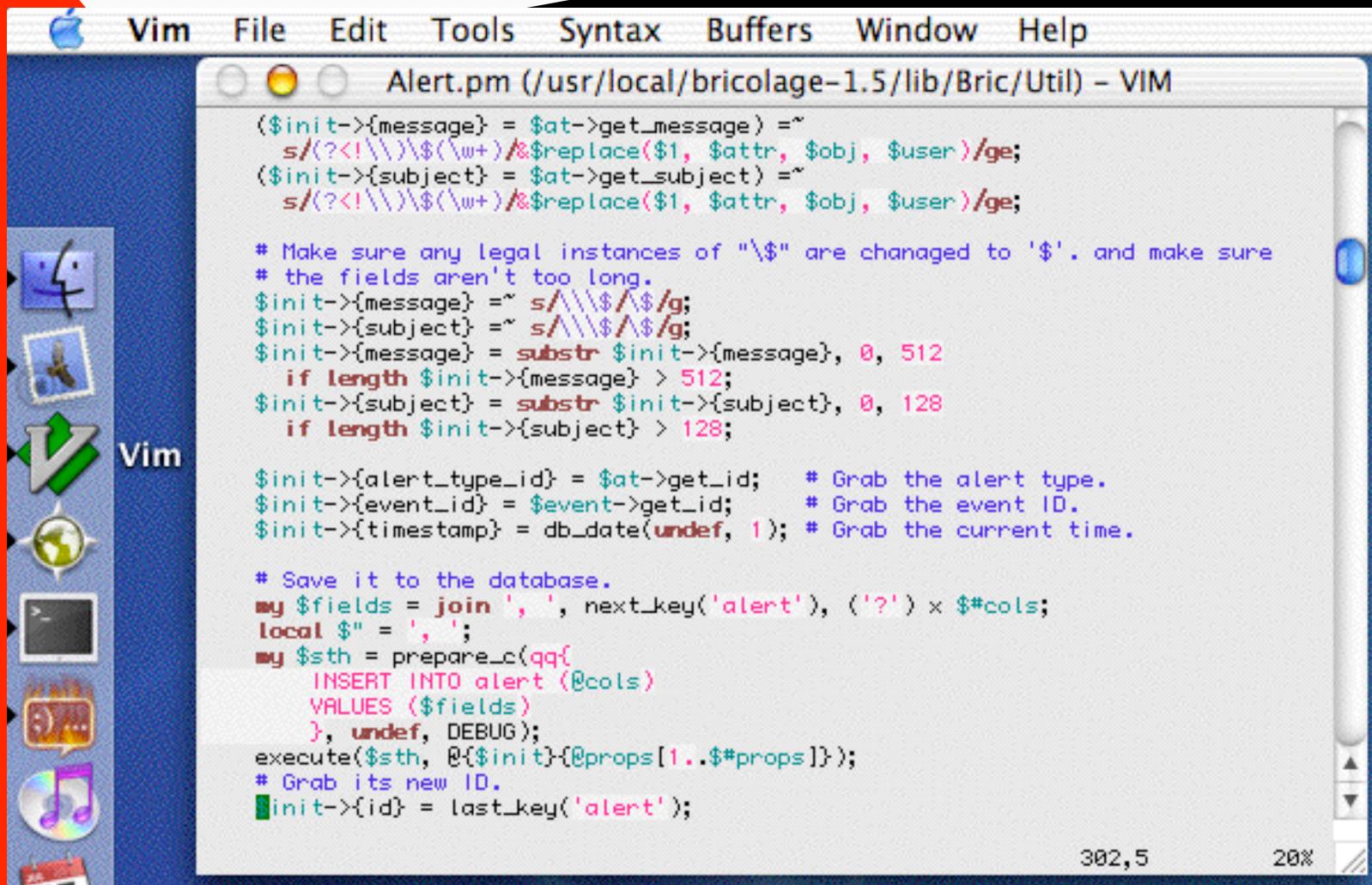
- <http://members.shaw.ca/akochi-emacs/>

Carbonized Vim

- Ported to Mac OS X by Dany St-Amant
- Binaries Available
- Sources Available at [ftp.vim.org](ftp://ftp.vim.org)



Carbonized Vim



A screenshot of the Carbonized Vim interface on a Mac OS X desktop. The window title is "Alert.pm (/usr/local/bricolage-1.5/lib/Bric/Util) - VIM". The code editor displays Perl code for an "Alert" module. The code includes regular expression substitutions for message and subject fields, handling of lengths, and database insertion logic using prepared statements. The Vim status bar at the bottom shows coordinates 302,5 and a zoom level of 20%.

```
$init->(message) = $at->get_message) =~
  s/(?!\$\$)\$\w+/$&replace($1, $attr, $obj, $user)/ge;
($init->(subject) = $at->get_subject) =~
  s/(?!\$\$)\$\w+/$&replace($1, $attr, $obj, $user)/ge;

# Make sure any legal instances of "\$" are changed to '$'. and make sure
# the fields aren't too long.
$init->(message) =~ s/\$\$/\$/g;
$init->(subject) =~ s/\$\$/\$/g;
$init->(message) = substr $init->(message), 0, 512
  if length $init->(message) > 512;
$init->(subject) = substr $init->(subject), 0, 128
  if length $init->(subject) > 128;

$init->(alert_type_id) = $at->get_id;    # Grab the alert type.
$init->(event_id) = $event->get_id;      # Grab the event ID.
$init->(timestamp) = db_date(undef, 1); # Grab the current time.

# Save it to the database.
my $fields = join ', ', next_key('alert'), ('?') x $#cols;
local $" = ', ';
my $sth = prepare_c(qq{
  INSERT INTO alert (@cols)
  VALUES ($fields)
}, undef, DEBUG);
execute($sth, @{$init}[@props[1..$#props]]);
# Grab its new ID.
$init->(id) = last_key('alert');
```

Carbonized Vim

- <http://macvim.swdev.org/OSX>

SSH Key Generation

- Create Public and Private Keys

```
% ssh-keygen -t rsa1  
% ssh-keygen -t rsa  
% ssh-keygen -t dsa
```

- Be sure to enter a passphrase for each

Using Public Keys

- Package up public keys

```
% cd ~/.ssh  
% cat identidy.pub > authorized_keys  
% cat id_dsa.pub > authorized_keys2  
% cat id_rsa.pub >> authorized_keys2  
% chmod 400 authorized_keys*  
% cd  
% tar zvcf .ssh/ssh-pub.tar.gz \  
.ssh/authorized_keys*
```

Using Public Keys

- Unpack ssh-pub.tar.gz on destination server(s)

```
% scp ~/.ssh/ssh-pub.tar.gz \
  destination.example.com:
% ssh destination.example.com
% tar zxvf ssh-pub.tar.gz
% chmod 755 .ssh
```

Passwordless SSH

- Set up ssh-agent and add the private keys

```
% eval `ssh-agent`  
% ssh-add
```
- Type in your passphrase(s) and you're ready to go!

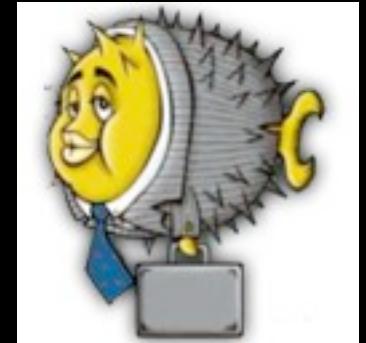
```
% ssh destination.example.com
```

Passwordless SSH

- Disadvantages to this Approach:
 - It only lasts as long as your terminal session.
 - Leaves ssh-agent hanging around
- Solution: Keep the ssh-agent environment variables persistent.

SSH Agent

- Easiest to use
- Native Carbon Application
- Stores Passphrase(s) in Key Chain
- Resource Hungry



<http://www.phil.uu.nl/~xges/ssh/>

SSHAgentServices

- Starts at Login
- Add private keys manually
- Best Solution
- Broken in Jaguar

<http://www-inst.eecs.berkeley.edu/~kevinvv/SSHAgentServices.html>

SSHAgentStartup

- Starts at System Startup
- All applications have access
- Add public keys manually
- Should continue to work

<http://www-inst.eecs.berkeley.edu/~kevinvv/>

<http://www.opendarwin.org/~kevin/SSHAgentStartup.tar.gz>

Apache/mod_perl

- Apache included with Mac OS X
- 100% Real Apache server
- mod_perl support via dyld
- Perl 5.6.0
- No support for Apache::Request and Apache::Cookie



Build Your Own!

- Build Perl 5.8.0
<http://developer.apple.com/internet/macosx/perl.html>
- Build Apache/mod_perl
 - Download Apache
 - Download mod_perl
 - Download libapreq Patch
 - Follow Instructions



Apache Links

- Apache

<http://www.apache.org/dist/httpd/>

- mod_perl

<http://perl.apache.org/dist/>

- libapreq

<http://www.apache.org/~joes/>

- My Instructions

<http://david.wheeler.net/osx.html>

PostgreSQL

- 7.2 and later Compile easily
- Create a “postgres” user
- Initialize the database
- Launch the Server
- Binaries available
- Resources



<http://www.entropy.ch/software/macosx/postgresql/>

<http://developer.apple.com/internet/macosx/osdb.html>

<http://www.macdevcenter.com/pub/a/mac/2002/06/07/postgresql.html>

MySQL

- 3.23 Compiles Well
- Create a “mysql” user
- Initialize the database
- Launch the Server
- Binaries available
- Resources



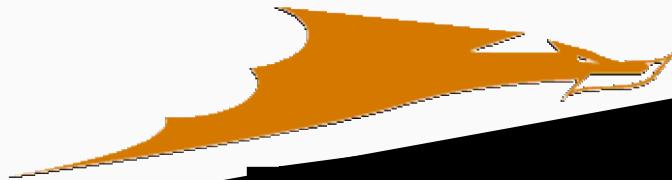
<http://www.entropy.ch/software/macosx/mysql/>
<http://developer.apple.com/internet/macosx/osdb.html>

X Windows

- XDarwin
<http://www.xdarwin.org/>
- OrobosX
<http://oroborosx.sf.net/>



Sendmail



- Included with Mac OS X
- You Must Enable it
- Instructions on MacDevCenter.com
 - Set DontBlameSendmail Option
 - Regenerate sendmail.cf
 - Set up NetInfo Settings
 - Enable Sendmail

Jaguar:

<http://www.macdevcenter.com/pub/a/mac/2002/09/10/sendmail.html>

10.1.5:

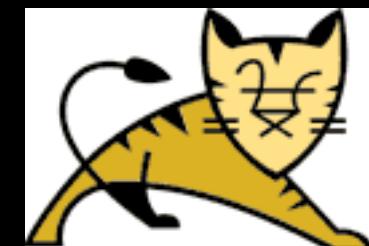
http://www.macdevcenter.com/pub/a/mac/2002/06/07/sendmail_1015.html

Java

- Java 1.3.1 Included
- Nice Integration with Cocoa APIs
- Nice Integration with Project Builder
- Nice Integration with Interface Builder

Tomcat

- It's Java, It's Mac OS X, Jakarta Tomcat just Works
- Can run Standalone
- Can run with Apache
- Apple Developer Article



<http://developer.apple.com/internet/java/tomcat1.html>



Ant

- It's Java, It's Mac OS X, Jakarta Ant just Works
- Grab it from the Jakarta Site
- Put it in /usr/local
- Symlink ant shell script in /usr/local/bin

<http://jakarta.apache.org/ant/index.html>

JUnit

- It's Java, It's Mac OS X, JUnit just Works
- Grab it from the Jarkarta Site
- Put it in /usr/local/ant/lib

<http://www.junit.org/>

Resources

- <http://www.macdevcenter.com>
- <http://developer.apple.com>
- <http://david.wheeler.net/macosx/>

Thank You

David Wheeler
Kineticode, Inc.
david@kineticode.com