

Open Source Project Promotion Case Study: CMake

Bill Hoffman

Kitware, Inc.

Bill.Hoffman@kitware.com



Changing the way “All” programmers build software: the CMake Project

- Introduction to CMake Features
- History of CMake
- Promoting CMake on the Web

CMake Features

- CMake = Cross Platform Make
- Open Source software construction tool
- Written in C++ and the CMake Language
- Uses native build tools
 - allow the developer to use what makes them most productive
 - make, nmake, Borland Make, watcom make
 - Visual Studio 6,7,8,9 IDE project files
 - XCode
 - KDevelop
 - Eclipse
- Platform inspection tool can search for
 - Programs
 - Libraries
 - Include paths
 - Determine hardware specifics like byte order



CMake Features (Cont.)

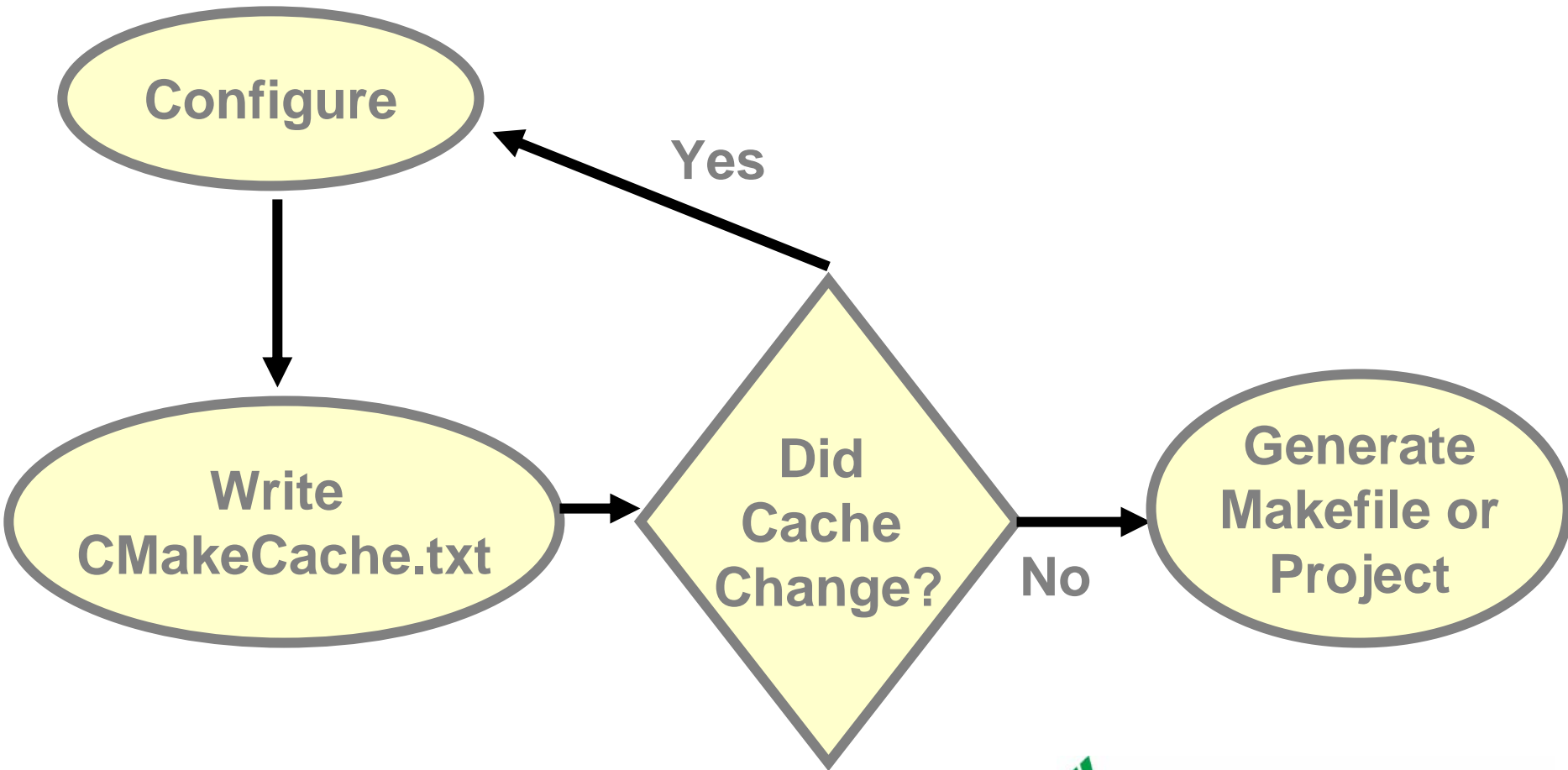
- Simple input file with simple language
 - Easy to do complex things:
 - `add_library(foo SHARED foo.cxx)`
 - Windows `dll`, Mac `.dylib`, HP-UX `.sl`, linux `.so`, UNIX `.so`
 - Hard stuff done in C++
 - Ideally it would be declarative, but pragmatically more language features are required. Still want to avoid programming too much in `CMakeLists.txt` files.

CMake Features (Cont.)

- Support for complex custom commands
 - VTK wrappers
 - QT's moc
- Optional component support
- Shared library and Dll building
- Create configured .h files, better than `-D` command line options
- Single input format for all platforms
- Automatic dependency generation
- Support for parallel builds on UNIX systems with a make that supports it
- Generate multiple Visual Studio projects from a single input file, useful for groups of test programs.
- Support for out of source build trees



CMake GUI (ccmake CMakeSetup) Process Flow



CMake History

- 1999 Start of the ITK project
- Kitware tasked with build environment
 - VTK current build system
 - autoconf on UNIX
 - pcmaker on Windows
 - Very specific to VTK and hard to maintain, no system introspection on windows



CMake History (Cont.)

- Requirements
 - Support UNIX/LINUX/Windows/MAC
 - Simple easy to use
 - Support code generation at build time (c++ wrapper support)
 - Do not require any software other than CMake to be installed.
 - Need only a C++ compiler (same as ITK itself)
 - Use Native Build tools like Visual Studio and Xcode



CMake History (Cont.)

Subject: [Insight-developers] CMake / ITK build process
From: Bill Hoffman <bill.hoffman@kitware.com>
Sender: insight-developers-admin@public.kitware.com
Date: 8/31/2000 6:29 PM
To: insight-Developers <insight-developers@public.kitware.com>

I have reworked the build process for ITK.
I have created a package called CMake (Cross platform make).
It is in Insight/CMake, and it includes all the makefile.in
and c++ source for the build process of ITK.

I have also updated and integrated the install documentation with the
InsightStart.doc in Insight/Documents. The InsightStart.doc
is also now available on the main ITK web page from the
top level (as a link):

<http://public.kitware.com/Insight/HTMLPrivate/>

Here is the direct link to InsightStart on the web:

<http://public.kitware.com/Insight/HTMLPrivate/InsightStart.html>

(BTW, You can use your cvs login and password to get into the private
web page area.)

The build process is almost the same as it was. However,
a few names have changed (pcbuilder -> CMakeSetup). Also, Makefile.in
is no longer used to list the files you want
built. The file used is CMakeLists.txt, which can no be found
in place of the old Makefile.in files.

As always, let me know if there are any problems.
I still have not fixed the Release build on windows.

-Bill

- ITK using precursors in Jan 2000
 - had to build ITK from the start
 - CMake born 8/31/2000



CMake History (Cont.)

VTK

Ken Martin ken.martin@kitware.com

Mon, 7 May 2001 16:05:00 -0400

Previous message: [\[vtk-developers\] Matrix Decomposition in vtkMath](#)

Next message: [\[vtk-developers\] New VTK directory structure and build process.](#)

Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

I have an experimental VTK build tree that uses a new make process (CMake) to build VTK on UNIX and windows. It also contains the new VTK directory structure. This is still at an early stage but you are welcome to take a look at it and provide feedback.

It can be obtained at

`cvs -d :pserver:you@www.visualizationtoolkit.org:/vtk/cvsroot co vtknew`

To build it there are (windows) directions in `vtknew/CMake/CMake.doc` For UNIX basically just do the following (edited as necc) `* ls * vtknew . . . * mkdir vtknew-linux * cd vtknew-linux * ./vtknew/configure * make` The build process is not complete right now but it should give you an idea of where we are heading. Note that all of the VTK files in this tree are the same as what is in the normal VTK tree. So if you edit `vtkObject.h` in the `vtknew` tree it will be edited on the main tree as well. In the CVS server they are symbolic links between the two trees so that changes in one tree show up in the other for all shared files (which is most of them) Ken



CMake History (Cont.)

- VTK
 - Early adoption, plenty of trouble
 - Caused some bad press

CMake History (Cont.)

- VXL 2001
 - Revision 1.1 - ([download](#)) ([annotate](#))
Mon Nov 6 08:48:51 2000 UTC (6 years, 11 months ago) by *remcvs_tgtjr_oxford*
Branch: **MAIN** beginning cvs CMake
 - 1 SUBDIRS = \
 - 2 v3p\
 - 3 vxl
 - 4 5 ME=TARGET
- ParaView, and all Kitware projects
- KDE 2006
- Open Scene Graph 2007



CMake Promotion

- Google alert
 - Watch for people talking about CMake on mailing lists and blogs
 - React to negative posts
 - Monitor CMake and CMake-promote mailing lists
 - Look for “ripe” projects
- Notable Projects we have tried to convert
 - 🥲 – wxWidgets (was wxWindows)
 - 👉 – KDE
 - 👉 – Scribus
 - 👉 – Open Scene Graph
 - ❓ – Boost

wxWindows

Re: Soliciting ideas for wxWindows 3.0

To: <wx-users@xxxxxxxxxxxxxxxxxxxxxx>

Subject: Re: Soliciting ideas for wxWindows 3.0

From: "Jolinda Smith" <jolinda@xxxxxxxxxxxxxxxxxxxxxx>

Date: Mon, 17 Mar 2003 11:46:15 -0800

What about using cmake (<http://www.cmake.org>) for this?

[wx-dev] New idea: Bakefiles in CVS From:

David Elliott <elliott@stcnet.com>

To: wx-dev@lists.wxwindows.org

Date: Jan 24 2004 - 2:54am



wxWidgets (Cont.)

- What went wrong?
 - did not like the fact that CMake had to be installed to do build.
 - Used to configure script being
 - Too early? (someday cmake will be preinstalled on all systems...)



KDE

9/21/2005 3:10 AM

Hi Bill,

> > Von: Bill Hoffman <bill.hoffman@kitware.com>

> >

> > Well, looks like we lost:

> > <http://dot.kde.org/1126452494/>

It seems so, but not yet 100% sure.

> > Thanks for all the effort. So what were the special targets that

> > caused the problems? Just wondering what we would have needed to add

> > to make the cut.

The current situation is the following:

There was the KDE conference in Malaga where I couldn't attend. So the developer of sconsbk/bkysys Thomas Nagy and the current KDE build system maintainer Stephan Kulow decided to use sconsbk. Not very surprising for me. Stephan likes Python and a Python-based solution might be easier to modify if special needs arise.

.....

I think the main reasons that sconsbk was more or less selected are:

-it uses Python -which Stephan really likes

-which is no "obscure" macro language

-which makes it easy to modify things, as opposed to the binary cmake

-on the first glance the cmake syntax (especially in upper-case) doesn't look nice

-there are no KDE projects yet, which use cmake, and there are already a few projects, which use sconsbk

Bye

Alex



KDE (Cont.)

1/10/2006 3:56 PM

Hi Brad, hi Bill,

here's the forwarded message:

----- Forwarded Message -----

Subject: Re: bksys / scones (Re: win32 port)

Date: Monday 09 January 2006 19:38

From: David Faure <faure@kde.org>

To: kde-buildsystem@kde.org

On Monday 09 January 2006 18:33, Jaison Lee wrote:

> > scones/bksys has been in development for months and yet it seems to be
> > months away at best from even being serviceable. Perhaps it's time to
> > cut our losses and run.

Honestly, that's my feeling too. As it is right now, we're stuck: no way to work on the configuration since it's not modular, and no way to compile kdelibs since kconfig_compiler can't be run.



KDE (Cont.)

1/10/2006 4:25 PM

Hi,

On Tuesday 10 January 2006 22:01, William A. Hoffman wrote:

- > > So, the best thing would be a working build that they could try.
- > > Are we better/worst than the current scons?
- > > What is your status right now?

IMHO it would make a really good impression if you would just post a small email to the list stating something like "here we are, the cmake developers, we'd like to support you, if you have questions, ask"

Bye
Alex



KDE (Cont.)

Jan 25th Alex announces that he can build kdelibs with CMake

1/20/2006 3:53 PM

Hi Bill,

On Monday 16 January 2006 04:33, you wrote:

> > Also, Alexander, let me know when this builds, and I will set up a
> > dashboard for it. Also, let me know how to get kde from cvs or svn.

this is the command to get current kdelibs from kde-svn:

```
svn co svn://anonsvn.kde.org/home/kde/trunk/KDE/kdelibs
```

Additionally you need some cmake stuff:

```
svn co svn://anonsvn.kde.org/home/kde/trunk/KDE/kdesdk/cmake
```

All cmake modules located there under cmake/kde4/ and cmake/modules/ have to go into the cmake Modules directory.

Ok, once you have all that, run cmake and hope that everything will be found.

Well, then make, and it should compile kdecop, kdeui, kwallet, kio, kdesu, kdefx and DCOP. The rest doesn't work yet.

Bye
Alex



KDE (Cont.)

2/13/2006 4:33 PM

> Von: Bill Hoffman <bill.hoffman@kitware.com>

> >

> > So, it seems that things are going well. I see a few grumbles here and
> > there, about bksys did it better, but things seem to be moving forward
> > well. What is your take on things?

I have the impression that we basically did it :-)

We have the OS X folks (i.e. Ranger Rick), we have some of the windows folks (these were the only ones which relied on scon) and we some Linux developers (me, Laurent Montel, very important David Faure).

Next week I'm on holidays, so there won't happen much.

> > Do we have any big features left? I would like to get an official

Release/debug build. It's on my todo.

> > CMake release out that can build kdelibs.

I'd wait a bit with that, until it is really official and also kdebase builds. I will start with this when I'm back.

It seems 2.3.3 doesn't have problems, so we can live with this for some time.

Bye
Alex



KDE (Cont.)

4/7/2006 2:58 AM

Hi Bill, hi Brad,

in preparation of an official announcement about KDE using cmake we'd like to get a short statement which we can include: Bill Hoffman said "...", and Brad King added "..."
Or something like this.

How should we refer to Kitware in one or two sentences ?

Bye
Alex

- Academy september 2006 irleand
 - Presented CMake and meet Alexander Neundorf
 - Open Source software - travel the world and meet interesting people, and not kill them..
 - Alex worked at Kitware for the past 6 months
- June 21, 2006 Why the KDE project switched to CMake -- and how linux world
 - <http://lwn.net/Articles/188693/>



Blog Power

- Conversations also show up on Blogs
- It is important to monitor Blogs and lists
- Use google alert
- Next slide shows example from KDE

KDE (Cont.)

SEARCH BLOG FLAG BLOG Next Blog» Create Blog

aseigo
the triumphs and travails of a shift-key-challenged KDE hacker

THURSDAY, APRIL 13, 2006

cmake sadness

my non-coding duties in kde had eaten into my coding time noticeably, but good stuff comes of it so it's a worthwhile use of time. the introduction of cmake, however, has redefined "waste of time" for me. today i get this:

```
Building CXX object libkonq/CMakeFiles/konq.dir/konq_popupmenu.o
/home/kde/kde4/kdebase/libkonq/konq_popupmenu.cc:21:18: error: qdir.h:
No such file or directory
/home/kde/kde4/kdebase/libkonq/konq_popupmenu.cc:23:19: error:
QPixmap: No such file or directory
/home/kde/kde4/kdebase/libkonq/konq_popupmenu.cc:24:21: error:
klocale.h: No such file or directory
/home/kde/kde4/kdebase/libkonq/konq_popupmenu.cc:25:26: error:
kapplication.h: No such file or directory
```

About Me



Aaron J. Seigo
[View my complete profile](#)

Twitter Updates

- Containment is going to rock. Groups of plasma applets with a background that can be pretty much anything: animated, opengl, interactive. [22 days ago](#)
- finding the coding mojo again; worked on the multiple timing stuff in dataengines. have to put it through the paces hard tomorrow before ... [27 days ago](#)
- good plasma meet on irc today. most of the code in playground has been figured

KDE (Cont.)



aseigo

the triumphs and travails of a shift-key-challenged KDE hacker

TUESDAY, NOVEMBER 21, 2006

kisses for cmake

the **cmake** team just announced the release of 2.4.4. seems like a good time to send some (free) kisses their way.

when we first started moving to cmake i kvetched about how much it was annoying me. since then, thanks in large part to alex and the cmake devs, it's come a long way (baby) and it's a dream to use.

faster than i ever remember autotools being. much less bloat in source packages. quite straightforward.

so ... kisses. =)



About Me



Aaron J. Seigo

 [View my complete profile](#)



Twitter Updates

- Containment is going to rock. Groups of plasma applets with a background that can be pretty much anything: animated, opengl, interactive. [22 days ago](#)
- finding the coding mojo again; worked on the multiple timing stuff in dataengines. have to put it through the paces hard

KDE (Cont.)

- Still Watching the KDE build list from time to time:

On Friday 05 October 2007, Brad King wrote:

> > David Faure wrote:

>> > > On Tuesday 02 October 2007, Brad King wrote:

>>> > >> There is alot of discussion in that bug report.

>>> > >>

>>> > >> The comparison of files is not for speed on installation...it's for not
>>> > >> changing timestamps of already installed files and causing all *other*
>>> > >> code using the installed files to rebuild, which is potentially even
>>> > >> slower.

>> >

>> > > Yes. The real solution is a dependency from the installed file to the file-to-install, so
>> > > that the file-to-install isn't installed if it isn't newer than the already-installed file.
>> > > And then, when that is implemented, there is no need for comparing files in order
>> > > to avoid changing timestamps, since you get for free by not installing the file at all :)

> >

> > This has now been implemented in CMake HEAD:

> >

> > <http://www.cmake.org/Bug/view.php?id=3349>

> >

> > Repeat installations are now lightning fast.

Thanks a lot, you rock!

I could see a huge speed improvement especially when working with the laptop on batteries, this is very nice.



KDE(Cont.)

sunday, october 07, 2007

CMake HEAD got faster reinstalls

On Friday Brad King announced on kde-buildsystem@kde.org "Repeat installations are now lightning fast", today i tried the CVS version of CMake (cmake version 2.5-20071007)

Well, it's not lightning fast, but it's a considerable win, here kdebase went from about 240 seconds to 75 seconds, so congrats to the CMake team and thanks for improving CMake in a continuous way!

posted by [albert astals cid](#) at [12:36 pm](#)

labels: [cmake](#)

2 comments:

anonymous said...

I think it's fair to say that repeat installs are now lightning fast. Before the install step make goes through all the directories and does... things... which (now) takes longer than the actual install step.

The best thing is that your system doesn't become unresponsive due to the surge of disk i/o when installing everything *again* as it was before.

Sunday, October 07, 2007 1:48:00 PM

pau garcia i quiles said...

I'm providing Ubuntu Feisty packages of daily CMake CVS checkouts. See <https://edge.launchpad.net/~pgquiles/+archive>

Sunday, October 07, 2007 2:58:00 PM

about me



Albert Astals Cid

I am a computer science graduate from Barcelona. I translate KDE to Catalan language and also am a proud

KDE developer.

[View my complete profile](#)

[KDE CODER](#) [KPDF](#)

search

Web This site

Earn money
by showing relevant
ads with **Google**
AdSense.

Ads by Google



Scribus

Home | About | Archives | Links | Contact



Scribus

Open Source Desktop Publishing

Scribus Developer Blog

Open Source Desktop Publishing

**Sat 20
May
2006** **Cmake Rocks**
Posted by mrdocs @ [Kudos](#)

Repeat after me:

Cmake Rocks

Cmake Rocks

Cmake Rocks

Archived Entry

Post Date :

Saturday, May 20th, 2006 at
11:37 am

Category :

Kudos

Do More :

You can leave a response, or
trackback from your own site.

Open Scene Graph

- Long time CMake user and OSG user Eric Wing promoted it on the OSG list
- I joined the OSG developers mailing list
 - Few posts in March 2007
 - Went well, no big battles, some confusion about in-source builds
 - We are implementing OSX framework creation abilities for them.



Boost ?

- The current battle is for Boost

David Abrahams wrote:

```
>
>
>> But perhaps we are using different definitions of "bigger". But as a
>> comparison some "real" number seem appropriate:
>>
>> * BB milestone 11 was downloaded about 37K times. Which would be an
>> indication of possible outside users. (Note, I can't find download
>> counts for Cmake, so perhaps Kitware can look those up and provide
>> them.)
>>
>
> Bill? That info would be most useful.
I do have some stats for that...
```

For the month of September we had the following downloads from the kitware server of cmake:

```
3000 windows binaries or about 100 a day
2501 source downloads about 83 a day
448 OSX universal binaries 13 a day
684 linux 22 a day
193 older versions of cmake 2.2.3 and 1.8.3 6 a day
```

So, that comes to 224 downloads per day. So, it would take CMake about 165 days for 37K downloads. I do not have running totals, but we have been going at a rate like this for some time now. I would guess that the total number of downloads exceeds 250K. Also, this also does not take into account the various linux distros that offer cmake binaries. I have no idea how many downloads they are getting.

-Bill



Lots of mailing lists!

The screenshot shows the Thunderbird email client interface. The left sidebar displays a list of mailing lists under the 'mailinglists' folder, including:

- angelfire
- boost (2105)
- boost-build
- cmake
- cmake-admin (266)
- cmake-bugs
- cmake-bugs-lookat
- cmake-bugs-tobe-closed
- cmake-commits (637)
- cmake-developers
- cmake-promote
- cygwin (9860)
- cygwin-apps (405)
- dart
- freecycle (2485)
- gcc-xml
- itk-developers (790)
- itk-users (818)
- itk-usersold
- kde-buildsystem (570)
- kde-core (8371)
- kde-dashboard (6123)
- mailinglist-owner-stuff (2309)
- make-w32 (295)
- mesa3d (1987)
- osg-build
- paraq (518)
- paraview (1929)
- valgrind (533)
- ve
- vtk-developers (313)
- vtkusers (4902)
- vxI-developers (54)
- vxI-users (258)

The main pane shows a list of messages with columns for Subject, Sender, and Date. The selected message is:

Subject: Re: [Boost-build] Future development
From: Bill Hoffman <bill.hoffman@kitware.com>
Date: 10/7/2007 9:46 AM
To: Boost.Build developer's and user's list

Vladimir Prus wrote:

why it makes any difference as long as other people with plenty of support resources are responsible for those 2.5M. We give away responsibility for maintaining 1.5M and take responsibility for 8000 lines of actual code written by Doug, most of which is comments.

Do you happen to know that KDE folks had to actually modify CMake? So at one point you could not build KDE by any released version of CMake? Repeat: KDE folks, not CMake developers.

OK, that is half true... The CMake developers did add a bunch of feature for KDE, and for some period of time, KDE could not use a standard release of CMake. However, I created interim releases and binaries for the KDE folks during this time.



Promotion on public lists

- Write the angry email, then delete it.
- Write the technology email and stick to the issues
- Stay cool!
- Ask a friend to read your post before posting

Lessons Learned

- OS Promotion is fun
 - Meet people from all over the world
 - Travel
 - Ireland, Belgium, UK, Finland
- What it takes for OS promotion
 - Constant watching (google alert)
 - A good team of developers
 - Pick your battles and features to add
 - Flexible funding, or lots of “free” time
 - Look for the low hanging fruit
 - Don’t sell it until it is ready
 - Direct development to needs that have the biggest payoff
 - Solve a problem, and solve it well

