## Subject: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Thu, 17 May 2007 15:35:21 GMT

I have an idea how to speed-up the porting (MacOS X now, be it is general).

The most time consuming part of problem is to find out all the information about implementing required things on target platform, something that developer that knows the platform would find primitive. OTOH, target platform guru's are unlikely to know about U++ implementation details.

So my idea is to create "reference application" that will contain all the function for minimal (and perhaps later, advanced) target platform support.

Target platform guru will reimplement this application (using the most straighforward way) and submit the code, which will serve as great boost to development speed (sort of U++ oriented knowledge base).

Thoughts?

Mirek

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by captainc on Sun, 14 Sep 2008 13:08:14 GMT

luzr wrote on Thu, 17 May 2007 11:35I have an idea how to speed-up the porting (MacOS X now, be it is general).

The most time consuming part of problem is to find out all the information about implementing required things on target platform, something that developer that knows the platform would find primitive. OTOH, target platform guru's are unlikely to know about U++ implementation details.

So my idea is to create "reference application" that will contain all the function for minimal (and perhaps later, advanced) target platform support.

Target platform guru will reimplement this application (using the most straighforward way) and submit the code, which will serve as great boost to development speed (sort of U++ oriented knowledge base).

Thoughts?

Mirek


I think this was a great idea. Was there any progress with it? What are we doing about Mac support? I think supporting Mac is necessary to get Mac developers to use and work on U++.

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Sun, 14 Sep 2008 15:37:50 GMT

View Forum Message <> Reply to Message

captainc wrote on Sun, 14 September 2008 09:08luzr wrote on Thu, 17 May 2007 11:35I have an idea how to speed-up the porting (MacOS X now, be it is general).

The most time consuming part of problem is to find out all the information about implementing required things on target platform, something that developer that knows the platform would find primitive. OTOH, target platform guru's are unlikely to know about U++ implementation details.

So my idea is to create "reference application" that will contain all the function for minimal (and perhaps later, advanced) target platform support.

Target platform guru will reimplement this application (using the most straighforward way) and submit the code, which will serve as great boost to development speed (sort of U++ oriented knowledge base).

Thoughts?

Mirek


I think this was a great idea. Was there any progress with it? What are we doing about Mac support? I think supporting Mac is necessary to get Mac developers to use and work on U++.


Unfortunately, I am afraid, for the time being, Mac stalled. I think Carbon EOL was the final hit...

Mirek

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by captainc on Tue, 16 Sep 2008 03:00:52 GMT

View Forum Message <> Reply to Message

Maybe go for a mac mini to test development for mac...

From apple store: $599.00
 http://store.apple.com/us/browse/home/shop_mac/family/mac_mi ni
From mac mall: $596.00
http://www.macmall.com/macmall/families/macmini_intel/

or even a used one.

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Tue, 16 Sep 2008 08:33:10 GMT
View Forum Message <> Reply to Message

Mac HW is the least problem here...


Mirek

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by bytefield on Tue, 16 Sep 2008 08:48:54 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 16 September 2008 11:33Mac HW is the least problem here...

Mirek
Then the lack of MacOS programmers? Seems no one is interested to port Upp to MacOSX... is that because of Objective-C or the upp core developers don't use Mac at all. I've never used a Mac and i don't know when i will use one, because i'm felling good using x86 platform with Linux and Windows. I know that MacOS is the second programs market after Windows(here in Europe, i don't know in USA) but i don't bother to make programs for it because MacOS have just a small amount of market.
So the question is, should we have interest in MacOS platform? Maybe others outside of Europe and better informed can give some hints.

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Tue, 16 Sep 2008 08:57:51 GMT
View Forum Message <> Reply to Message

bytefield wrote on Tue, 16 September 2008 04:48luzr wrote on Tue, 16 September 2008 11:33Mac HW is the least problem here...

Mirek
Then the lack of MacOS programmers?


IMO, lack of Core U++ developers as well....

The problem is that this requires both deep MacOSX knowledge AND deep U++ knowledge...

Quote:
Seems no one is interested to port Upp to MacOSX... is that because of Objective-C or the upp core developers don't use Mac at all.


IMO, both...

---

Quote:
I know that MacOS is the second programs market after Windows(here in Europe, i don't know in USA)


In my country, there is IMO much more Linux related programming *jobs* than MacOS programming jobs. Basically, any ISP needs Linux programmers. Who needs OSX coders?

MacOSX is used by graphics, but they do not need any software developemnt...

AFAIK, in USA situation mich be much different, Apple has much deeper market penetration.

Quote:
So the question is, should we have interest in MacOS platform?

Definitely. But other things have priority now....

Mirek

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by cbpporter on Tue, 16 Sep 2008 09:06:27 GMT
View Forum Message <> Reply to Message

bytefield wrote on Tue, 16 September 2008 11:48
Then the lack of MacOS programmers? Seems no one is interested to port Upp to MacOSX... is that because of Objective-C or the upp core developers don't use Mac at all. I've never used a Mac and i don't know when i will use one, because i'm felling good using x86 platform with Linux and Windows. I know that MacOS is the second programs market after Windows(here in Europe, i don't know in USA) but i don't bother to make programs for it because MacOS have just a small amount of market.
So the question is, should we have interest in MacOS platform? Maybe others outside of Europe and better informed can give some hints.
I think that it is about the lack of MacOS developers. Objective-C si not that hard, but still, you can't just read a little about it and expect to write anything meaningful. You need some experience, and I think that a lot of people would rather avoid having to learn a new and relatively useless (except for Mac) programing language.

And even if you know ObjectiveC, it will take some time before you get a basic window and message system running, and you also must be able to integrate it into CtrlCore. That's why it would be the best if someone who knows Cocoa and someone familiar with CtrlLib implementation work together. So basically one does a straightforward Cocoa minimal application, like you would find in a tutorial, and the other then tries to adapt if for CtrlLib.

Too bad that Carbon is outdated .

And of course these people need Macs. Anybody successfully emulated a Mac?

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by bytefield on Tue, 16 Sep 2008 09:44:52 GMT

luzr wroteIn my country, there is IMO much more Linux related programming *jobs* than MacOS programming jobs. Basically, any ISP needs Linux programmers. Who needs OSX coders? That's happen also in my country and I guess all around the world, IMO LAMP(Linux+Apache+MySQL+Php) is one of the best solutions for server side applications, because it's cheap and opened.

When i was talking about the MacOS as a second market after Windows i was thinking that Linux market side want mostly open-source programs and just the best oss get sales in Linux market(maybe not source or program sales but support sales). BTW, have someone from here made a commercial application for Linux and/or open-source/closed-source in the same time?

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by gprentice on Tue, 16 Sep 2008 12:25:32 GMT

Maybe it's possible to do a version of upp that sits on top of factor and doesn't directly know anything about the underlying platform.  Not volunteering though.

http://factorcode.org/

Graeme

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by captainc on Tue, 16 Sep 2008 13:06:25 GMT

Quote:
Then the lack of MacOS programmers?

IMO, lack of Core U++ developers as well....
The problem is that this requires both deep MacOSX knowledge AND deep U++ knowledge...
Things look like they are ramping up for U++. Any options for another Core developer?
Have you investigated cloning?! A second Mirek would help...!

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by guido on Tue, 16 Sep 2008 15:49:08 GMT

I guess OS X market is end-user focused, while most programming is really done in vertical markets, in-house (Java) apps etc. or web-related.

Maybe applying for Google Summer of Code is worth a shot next year. No idea what chances there, though.

---

Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by captainc on Thu, 18 Sep 2008 00:18:25 GMT
View Forum Message <> Reply to Message

New product that I saw from a Digg article:
http://www.efixusa.net/product_info.php?products_id=28

Its a usb drive that allows you to install MAC OS on a pc
From their site:
Quote:Run Unmodified Mac OS X on a PC:
Boost your creativity, your dreams and daily tasks in a fully new dimension with Mac OS X on PC.
Start to experience what Mac OS X users always enjoyed, stability of Mac OS X system, unique security protection, no viruses, and the beauty of the OS all on your standard PC.
Note sure how it works yet...
Update: Looks like it is limited to the hardware it can run on.

---

Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Thu, 18 Sep 2008 16:24:27 GMT
View Forum Message <> Reply to Message

captainc wrote on Wed, 17 September 2008 20:18New product that I saw from a Digg article:
http://www.efixusa.net/product_info.php?products_id=28

Its a usb drive that allows you to install MAC OS on a pc

From their site:
Quote:Run Unmodified Mac OS X on a PC:
Boost your creativity, your dreams and daily tasks in a fully new dimension with Mac OS X on PC.
Start to experience what Mac OS X users always enjoyed, stability of Mac OS X system, unique security protection, no viruses, and the beauty of the OS all on your standard PC.
Note sure how it works yet...
Update: Looks like it is limited to the hardware it can run on.

Who cares about "unmodified"

(It is definitely possible and not even that much hard to run OSX on generic PC hardware, even on AMD CPUs. Of course, it is not legal too....)

Mirek

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by jeremy_c on Thu, 05 Nov 2009 14:53:14 GMT
View Forum Message <> Reply to Message

No more word on OS X support I guess? To me, that is the one thing holding me back going full force with U++. I would like to deploy on Windows, Linux and OS X. I am betting it's the case with many other developers as well. Here in the USA, OS X is here to stay and we really must think about deployment on it :-/

Jeremy

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by Reini on Fri, 06 Nov 2009 00:11:55 GMT
View Forum Message <> Reply to Message

Hello OS X Fans,

I am looking forward in buying a Mac and learning to code for it. Ok I am no expert on Mac as Mirek is definitely right we should have for OS X porting. At the moment I am just lerning U++ and will contribute some stuff in the near future.

So I was also thinking to do some Mac Programming. But as I realized that the native language is there Objective-C. I was wondering how to port then something in U++ ?

But for my understanding of the OS X Architecture its a Unix based system so C++ should be definitely possible to use even if only with a binding from Objective-C to C++.

But question is if that is fast enough for OS X ?

With some help of you I would be very interesting in porting U++ to OS X and I am sure if someone does the first step others will follow

Greetz Reini

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Sat, 07 Nov 2009 05:38:16 GMT
View Forum Message <> Reply to Message

Reini wrote on Thu, 05 November 2009 19:11Hello OS X Fans,

I am looking forward in buying a Mac and learning to code for it. Ok I am no expert on Mac as Mirek is definitely right we should have for OS X porting. At the moment I am just lerning U++ and will contribute some stuff in the near future.

So I was also thinking to do some Mac Programming. But as I realized that the native language is there Objective-C. I was wondering how to port then something in U++ ?

But for my understanding of the OS X Architecture its a Unix based system so C++ should be definitely possible to use even if only with a binding from Objective-C to C++.


AFAIK, you can use Objective-C++ as well. I guess it should be possible to encapsulete Cocoa API so that there is no "Objective" left outside the encapsulation.

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by koldo on Sat, 07 Nov 2009 12:44:01 GMT
View Forum Message <> Reply to Message

Hello all

I do not know about Mac programming, but this sound strange to me:
Quote:So I was also thinking to do some Mac Programming. But as I realized that the native language is there Objective-C. I was wondering how to port then something in U++ ?

So just searching a little bit I have found that in Mac it is used Xcode Tools (http://developer.apple.com/tools/), previously named Apple DevTools, including Mac versions of gcc, gdb and make, here http://developer.apple.com/tools/gcc_overview.html

Best regards
Koldo

PD. Fixed gcc link thanks to Andrei Natanael

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by andrei_natanael on Sat, 07 Nov 2009 20:11:54 GMT
View Forum Message <> Reply to Message

koldo wrote on Sat, 07 November 2009 14:44Hello all

I do not know about Mac programming, but this sound strange to me:
Quote:So I was also thinking to do some Mac Programming. But as I realized that the native language is there Objective-C. I was wondering how to port then something in U++ ?

So just searching a little bit I have found that in Mac it is used Xcode Tools (http://developer.apple.com/tools/), previously named Apple DevTools, including Mac versions of gcc, gdb and make, here http://developer.apple.com/tools/gcc_overview.html

Best regards
Koldo
Hi Koldo, i'm not a Mac programmer but recently i've looked over programming topics for mac on internet (especially apple site). There was possible to write GUI in C/C++ for MacOS using

Carbon library but Apple dropped it and new library Cocoa is only Objective-C. It may sound strange but Objective-C is really used to do programs for macs

I know you can mix C/C++ with Objective-C/C++ and up to some point Objective-C is pure C, then the differences comes. We have to use Objective-C mixed with C/C++ to create an interface which could get used by upp CtrlCore. Btw, we may get some inspiration by looking at Qt code .

This year(2009-2010) i will be a graduated student, Bachelor of Mathematics-Computer Science (i hope i wrote this correct) and my thesis is about cross-platform GUI and i'm writing a Chameleon like library for Windows, Linux(KDE or Gnome or both) so if i'm not failing at doing that than i will try to port it to macos too after exams and if it's good enough maybe upp will adopt it

P.S.: your last link should point to http://developer.apple.com/tools/gcc_overview.html (without dot at the end of the link)

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by Reini on Sat, 07 Nov 2009 21:32:42 GMT
View Forum Message <> Reply to Message

Hi Guys,

Finally I did it today and baught a Mac Mini
It was a special offer and I got also a printer and an Ipod Nano for free

And I have to admit that the feeling to work on Mac is for daily office use a real pleasure.

Now I have to investigate the system a bit deeper.
Hot to find the console and such real basic stuff.

I have installed already Opera and found this Xcode stuff to be the only ? solution that is best at the moment. Codewarrior was for a long time the leading C++ Environment for Mac but is now baught by Freescale and so I think not sold any more. But anyway even Eclipse with CDT should do the job.

I will try to get involved in the development details and tell you further on the steps.

Mirek would you like to open a separate Forum for Mac or so ? Or are you developers also good reachable via IM ? This would be my preferable way of communicating since the small steps would flood then the forum too much

So far have a nice weekend

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by koldo on Sat, 07 Nov 2009 21:47:39 GMT
View Forum Message <> Reply to Message

Thanks Andrei Natanel and all

Just a link about it:  http://stackoverflow.com/questions/525609/use-c-with-cocoa-instead-of-objective-c

The forced mix seems to be Core in C++ & GUI in Objective-C + Cocoa.

Best regards
Koldo

---

Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by andrei_natanael on Sun, 08 Nov 2009 06:16:22 GMT
View Forum Message <> Reply to Message

Reini wrote on Sat, 07 November 2009 23:32Hi Guys,
Mirek would you like to open a separate Forum for Mac or so ? Or are you developers also good reachable via IM ? This would be my preferable way of communicating since the small steps would flood then the forum too much

You'll find Mirek ICQ address in /uppbox/developers.txt in svn repo. Other developers have ICQ too and you'll find their addresses in forum profile for each of them. (i personally dislike ICQ, it is too tied to their client, i.e. you cannot change your password from other client or even icq web site and their client is windows only).

I would like discussions to be in forum because in that way we(especially me) will learn about porting a piece of software, mixing languages, MacOSX architecture etc. and also get informed about porting progress.

---

Subject: Suggestion
Posted by Reini on Fri, 13 Nov 2009 19:58:22 GMT
View Forum Message <> Reply to Message

Hello Mirek and others,

I just found an deprecated C++ Programming Framework called nano.
Probably we can take some ideas from there ?

Otherwise I would suggest just to start at first with the Core of U++ and start to port that stuff step by step to OSX ?

From the current wxWidgets Project I think we can borrow also a bit knowloedge

In the meantime I am playing a bit around with XCode the official Apple Developer Tools.

## Subject: Re: Suggestion
Posted by andrei_natanael on Fri, 13 Nov 2009 20:23:47 GMT

View Forum Message <> Reply to Message

Reini wrote on Fri, 13 November 2009 21:58
I just found an deprecated C++ Programming Framework called nano.
Probably we can take some ideas from there ?
Your link is not visible because you used html code
Nano Framework

## Subject: Re: Suggestion
Posted by mirek on Sat, 14 Nov 2009 04:41:21 GMT

View Forum Message <> Reply to Message

Reini wrote on Fri, 13 November 2009 14:58Hello Mirek and others,

I just found an deprecated C++ Programming Framework called nano.
Probably we can take some ideas from there ?

<a href="http://www.refnum.com/software/nano/index.html" target="_blank"></a>

Otherwise I would suggest just to start at first with the Core of U++ and start to port that stuff step by step to OSX ?

From the current wxWidgets Project I think we can borrow also a bit knowloedge

In the meantime I am playing a bit around with XCode the official Apple Developer Tools.

Actually, Core is the least problem, it will compile almost right away - OSX is basically BSD on that level, which is well supported.

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by Reini on Sat, 21 Nov 2009 17:43:02 GMT

View Forum Message <> Reply to Message

Hello all,

I just checked a bit the XCode Development platform for Mac and realized that they follow the Model-View-Controler Pattern.

So the question is for me if it would be possible to put all in MVC Style to U++ or if its using also the MVC Pattern ?

Greetz

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Sun, 22 Nov 2009 22:05:45 GMT
View Forum Message <> Reply to Message

Reini wrote on Sat, 21 November 2009 12:43Hello all,

I just checked a bit the XCode Development platform for Mac and realized that they follow the Model-View-Controler Pattern.

So the question is for me if it would be possible to put all in MVC Style to U++ or if its using also the MVC Pattern ?

Greetz

No, but I think it would still be possible to do Cocoa U++.

Each window will simply consist of sigle view and MVC will implement U++ CtrlCore API. Obviously, using as little from Cocoa as possible (same is true for Win32 and X11).

Mirek

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by kohait00 on Thu, 26 Aug 2010 09:03:45 GMT
View Forum Message <> Reply to Message

there is a core layer, called 'quartz' which could be used, without the need to have objective c++, trying to find out what we need for..but we wouldnt need cocoa..

Quote:
Introduction

Quartz 2D is an advanced, two-dimensional drawing engine available for iOS application development and to all Mac OS X application environments outside of the kernel. Quartz 2D provides low-level, lightweight 2D rendering with unmatched output fidelity regardless of display or

the final destination when you use the Quartz 2D application programming interface (API) for drawing.

The Quartz 2D API is easy to use and provides access to powerful features such as transparency layers, path-based drawing, offscreen rendering, advanced color management, anti-aliased rendering, and PDF document creation, display, and parsing.

The Quartz 2D API is part of the Core Graphics framework, so you may see Quartz referred to as Core Graphics or, simply, CG.

 http://developer.apple.com/mac/library/documentation/Graphic

sImaging/Conceptual/drawingwithquartz2d/Introduction/Introdu ction.html

and: it's not to be mixed with QuickDraw API, which *is* deprecated, but quartz is a replacement
for it.. cocoa partly uses it as far as i know, but mainly it uses opengl directly..

i have a test program here drawing stuff on screen with quartz, my chief bought a mac shortly, we
are experimenting and really considering porting.

anyone with infos on it please post..

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by kohait00 on Thu, 26 Aug 2010 09:41:30 GMT
View Forum Message <> Reply to Message

i googled a bit..

 http://www.iphonedevsdk.com/forum/iphone-sdk-development/352 9-best-api-2d-game.html

look for 'quartz' here
http://www.markj.net/iphone-development-programming-books/

http://developer.apple.com/technologies/mac/cocoa.html

 http://stackoverflow.com/questions/1736583/quartz-2d-vs-open gl-es-learning-curve
states quartz is a 'c api'

 http://my.safaribooksonline.com/0321336631/9#X2ludGVybmFsX0Z
sYXNoUmVhZGVyP3htbGlkPTAzMjEzMzY2MzEvMTk=
has got some excerpt info on API parts

Quote:
Quartz 2D is a part of the Core Graphics system. It is a modern graphics library based on the
imaging
model that Adobe created originally for PostScript printers and later as part of the PDF graphics
file
format. This is the same imaging model that graphics professionals have used for several years to
create the artwork on everything from books and advertisements to application splash screens
and
on-line games.
The Quartz 2D drawing model allows you to create sophisticated graphics with a simple API. The
Quartz 2D imaging model is quite different from the drawing models of other graphics libraries. Its
library can draw to many kinds of graphics devices while maximizing the fidelity of the graphics on
each device. As a result, developers familiar other graphics libraries such as QuickDraw, GDI
from

work

---

with the device and resolution independent drawing model in Quartz 2D.


Quote:
The methods that you call on Core Graphics objects in Python take their names from the routines in the C API that invoke the same behavior.


i conclude that Quartz itself is C API, with wrappers to several languages i.e objective c++, cocoa and carbon both use quartz for some parts. (carbin more than cocoa)

so a mac application would be a thin objective c++ application layer having almost all parts in c / c++

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by kohait00 on Thu, 26 Aug 2010 09:48:04 GMT
View Forum Message <> Reply to Message

and some more

Quote:
The Quartz 2D C API

Quartz 2D is one part of a larger graphics environment known as Core Graphics. In addition to Quartz 2D, Core Graphics interacts with the window system and hardware to integrate graphics created by the many different graphics libraries on the system. It is responsible for managing the display environment and working with the window server and Quartz compositor to create the image presented on the screens. While Core Graphics encompasses more than Quartz 2D, it's not unusual for people to use the two terms interchangeably when discussing Quartz 2D in context.
The Core Graphics API is an object oriented API. To allow developers to use that API in as many environments as possible, however, the system implements it as a set of C routines. The API itself draws upon the object-oriented concepts of encapsulation, inheritance, and polymorphism. Because C doesn't directly support these concepts, Core Graphics uses opaque data types, in the place of classes, to support encapsulation and inheritance. To provide polymorphism, Core Graphics applies a naming convention to its routines.

http://flylib.com/books.php?ln=en&n=3&p=310&c=9& amp;p1=1&c1=1&c2=100&view=1

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by jeremy_c on Sun, 29 Aug 2010 00:36:04 GMT
View Forum Message <> Reply to Message

Is anyone working on an OS X port? OS X is a major platform now. When comparing U++ to QT

and wxWidgets, lack of OS X support will be a negative toward U++.

Jeremy

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by fudadmin on Thu, 11 Nov 2010 10:56:28 GMT

View Forum Message <> Reply to Message

I moved this topic here from "U++ developers corner" partly because:

luzr wrote on Thu, 17 May 2007 16:35I have an idea how to speed-up the porting (MacOS X now, be it is general).

The most time consuming part of problem is to find out all the information about implementing required things on target platform,
[...]

Thoughts?

Mirek

And this topic contains some useful info. Maybe it would be a good idea to create a sticky topic containing a concentrated, most relevant objective-c++ snippets collection with info and links aka "U++ porting aqua resources"?

jeremy_c wrote on Sun, 29 August 2010 01:36Is anyone working on an OS X port? [...]

Jeremy

Regarding "reference application". Maybe a good idea would be to start with upp SystemDraw (the first thing doesn't compile with Xcode) using Quartz/CoreGraphics CGContextRef (easier and gives clipping rectangles capabilities) or Cocoa NSGraphicsContext (need info about clipping rectangles...)?

P.S ConsoleDraw (Draw, Painter, PdfDraw ...) compiles with Xcode and works.

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Sun, 21 Nov 2010 17:03:58 GMT

View Forum Message <> Reply to Message

fudadmin wrote on Thu, 11 November 2010 05:56
Regarding "reference application". Maybe a good idea would be to start with upp SystemDraw

---

(the first thing doesn't compile with Xcode) using Quartz/CoreGraphics CGContextRef (easier and gives clipping rectangles capabilities) or Cocoa NSGraphicsContext (need info about clipping rectangles...)?


Definitely.

Quote:
P.S ConsoleDraw (Draw, Painter, PdfDraw ...) compiles with Xcode and works.

Well, good news there!

Mirek

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by kohait00 on Tue, 05 Jul 2011 14:41:27 GMT
<>

seems as if Quartz 2D has similar drawing mechanisms like Upp. id defines a Drawing context, which finally is rasterized when the final resolution is available. so, porting *should* be generally not too complicated.

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Tue, 05 Jul 2011 19:42:51 GMT
<>

kohait00 wrote on Tue, 05 July 2011 10:41seems as if Quartz 2D has similar drawing mechanisms like Upp. id defines a Drawing context, which finally is rasterized when the final resolution is available. so, porting *should* be generally not too complicated.

..besides, you only need to implement DrawRect, DrawText and DrawImage to run e.g. theide...

Mirek

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by kohait00 on Tue, 05 Jul 2011 20:17:45 GMT
<>

Who,s willing to start the experiment? Imagine the kind of boost for ultimate beeing able to run natively on win32/posix/macos/sdl/framebuffer...

This can be a key feature especially for embedded systems, it,s quite hard to start with qt, fltk is outdated and pretty ugly, picogui has ceased devolpment, xserver or fbcon are complicated to implement in ES, and here comes upp, one tool for all the needs..even adaptable to a custom

backend..

EDIT: this one is a central source..
 http://developer.apple.com/library/mac/#documentation/Graphi
csImaging/Conceptual/drawingwithquartz2d/Introduction/Introd uction.html

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Wed, 06 Jul 2011 10:34:12 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 05 July 2011 16:17Who,s willing to start the experiment?

Experiment? I guess we are past that now. Now we simply need to develop MacOS X backend....

Anyway, I hope somebody else than me will uptake the task.

Mirek

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by daveremba on Mon, 11 Jul 2011 04:52:46 GMT
View Forum Message <> Reply to Message

Hi,  I'm new to UPP, so I'll pose some
basic questions for discussion below.

I have some time and knowledge of the MacOS,
C++, etc. and would consider helping with
a more complete port of UPP to Mac.
It would help to get some direction/discussion
from the UPP community.
Also, are there any other developers on
MacOSX who would work on this??

What is the interest level in a MacOS port
of UPP?


I got a port working of TheIDE on MacOSX,
and the example apps run, but under an X11
emulator, and do not have the Apple look and feel.
This of course is expected.

There is some remaining work to fix a couple of
problems with buttons and menus.  But in any

case this is a quick and crude port, and its main
value is that TheIDE itself does run and could
be used as an alternative to Xcode.

Xcode is the Apple equivalent of MS Visual Studio,
or KDevelop, etc. It generates code for MacOS,
iPhone, and iPad.  It has a simulator for
the mobile devices (like Android).
One works mainly in Objective-C.

Xcode generates a .nib file, similar to the UPP
.layfile, form another tool called "Interface
Builder".

Attached are snapshots of the "widgets" available
in Xcode IB for iPhone and MacOSX.

Some questions to consider:

What kind of port would be reasonable and useful
to the UPP community?
Probably reinventing
the entire Apple tool chain is not desirable.

is there a 1:1 correspondence of widgets to UPP?
(probably not)

The UPP community would want an app that runs
on Linux and Windows to look similar, but also
conform to Apple look and feel, right?
(without much or any re-coding effort, like
Qt etc, but a better result than Qt)

There are several technical paths to chose from;
and picking the best one depends on what
the community expects from a porting effort
of UPP to Mac.

So a useful first step, I would ask readers to please
look at the attached widgets, and provide some
comments as to which ought to be connected to UPP,
and which are "out of scope" (at least for a first
port).   (widgets are in messages that follow)

There is at least a basic correspondence of
menus, buttons, tabs, etc. There are quite a
few button styles on MacOS however.
You can also

see some specialized containers that Apple provides
for OpenGL, Quicktime, PDF, and choosers for
time, date, options, etc. (particularly for
iPhone they look quite different from UPP).

I saw that another developer, Aris (fudadmin), did a port
of the vector and image drawing tools in UPP.
I downloaded that Xcode project also.


off-topic:
A universal mobile UI framework that claims 1 million
developers for mobile devices and that I'm working
with is Sencha, which runs in Javascript.

The libarary is called ExtJS.  Here is the
link:
http://www.sencha.com/products/extjs/

Sencha currently lacks a good data-object
connection to server (Direct-EXT is incomplete).
I recommend UPP developers take a look at it,
how might it relate to the Rainbow (or future UPP)
projects?

For another project, I am considering using UPP for the
server backend (since it is so fast compared to PHP, and
a much better language (C++ & good design)
compared to PHP).  I'd need JSON/REST in UPP
to get my project working however.

Any comments on this would be appreciated
(maybe in another thread).

Thanks,

Dave

Xcode Interface builder widgets (Cocoa library)
are attached as 4 replies to this post.

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by daveremba on Mon, 11 Jul 2011 04:54:48 GMT
View Forum Message <> Reply to Message

screen 1

```
File Attachments
```

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by daveremba on Mon, 11 Jul 2011 04:56:18 GMT
View Forum Message <> Reply to Message

screen 2

```
File Attachments
```

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by daveremba on Mon, 11 Jul 2011 04:57:43 GMT
View Forum Message <> Reply to Message

screen 3

```
File Attachments
```

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by daveremba on Mon, 11 Jul 2011 04:59:14 GMT
View Forum Message <> Reply to Message

iphone widgets

```
File Attachments
```

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by kohait00 on Mon, 11 Jul 2011 07:34:34 GMT
View Forum Message <> Reply to Message

thanks for providing some insight..

the level of porting upp would be 'native', grabbing a spot of screen or a main window, and beeing able to draw on it. this can start off with the X11 based theIDE port, but it should end with a full rainbow complaint (see uppsrc/rainbow) port of the underlying structure. everything in between is sort of implementation detail

concerning the widgets: upp has its own set of wifgets, so we don't need to mess around with the macos stuff at all, this is also the beauty of crossplatform.. an almost-macosx style of the upp widgets can be obtained with the skinning mechanism in upp. you already are able to have macos style in upp apps in windows..

so the basic thing is:

* implement the SystemDraw interface, which redirects the corresponding drawing operations (like DrawRect, DrawLine) to the native drawing contexts of macos.

* implement the mouse processing to translate the received mouse handling from macos to upp and have upp process it.

* implement the keyboard processing, translate the received key actions from macos to upp and have upp process it.

* when this quite works, stuff like copy/paste, drag/drop etc. can be handled (with macos means).

i'm novice in mac stuff (actually have done nothing with it, besides some research in the web for quartz port). but i will try to help. since MACOSX port of UPP would be a MAJOR boost for upp, and our company would HIGHLY apriciate, we have audio controlling software coded in upp, customers have windows, linux, AND ofcorse macos, and need to run it in simulaters. i'm a bit into the porting stuff (not too much though), porting to linux native framebuffer and SDL. so the process is comparable. but it this case, you have main windows of macos (which should map to TopWindow), and i have a single sheer of buffer which is esier..

---

## Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Mon, 11 Jul 2011 12:59:11 GMT
View Forum Message <> Reply to Message

daveremba wrote on Mon, 11 July 2011 00:52
What kind of port would be reasonable and useful
to the UPP community?
Probably reinventing
the entire Apple tool chain is not desirable.


I always tend to answer myself: Firefox.

Despite looks, Firefox is emulated GUI on Win32 and Linux, just like U++. I believe it is emulated

GUI on MacOSX to, hint is e.g. here:

 http://mxr.mozilla.org/firefox/source/widget/src/cocoa/nsNat iveThemeCocoa.h

Quote:
is there a 1:1 correspondence of widgets to UPP?
(probably not)


Well, the desired result is that any single existing example in examples and reference runs on MacOSX in native mode and feels native. We are not aiming at 100% look&feel. 97% is just enough

Quote:
The UPP community would want an app that runs
on Linux and Windows to look similar, but also
conform to Apple look and feel, right?
(without much or any re-coding effort, like
Qt etc, but a better result than Qt)


Exactly. And we already know it is possible Linux vs Windows wise.

Quote:
There are several technical paths to chose from;
and picking the best one depends on what
the community expects from a porting effort
of UPP to Mac.


Actually, I do not think so. There is only one possible path and it is the same as current Linux and Windows backends...

I have outlined steps in previous posts.

Just to make things clear, we will NOT be using Cocoa widgets. We will be emulating them.

So the whole xcode/nib thing is mostly irrelevant to us. We will only use as little so that MacOSX recognizes our app as native. Nothing more.

To MacOSX it will be just windows with simple view and no subwidgets...


Quote:
off-topic:
A universal mobile UI framework that claims 1 million
developers for mobile devices and that I'm working
with is Sencha, which runs in Javascript.

The libarary is called ExtJS.  Here is the
link:
http://www.sencha.com/products/extjs/

Sencha currently lacks a good data-object
connection to server (Direct-EXT is incomplete).
I recommend UPP developers take a look at it,
how might it relate to the Rainbow (or future UPP)
projects?

For another project, I am considering using UPP for the
server backend (since it is so fast compared to PHP, and
a much better language (C++ & good design)
compared to PHP).  I'd need JSON/REST in UPP
to get my project working however.

Any comments on this would be appreciated
(maybe in another thread).

Interesting. I see rainbow and MacOSX as our responsibility, something that simply needs to be
done, but after that, I definitely want to work quite a lot on U++ as web framework. Maybe even
embrace Javascript in theide in some way...

That said, U++ is already being used as backend quite intensely in my work. But we are still
missing "higher level" web framework...

Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by mirek on Mon, 11 Jul 2011 13:06:14 GMT
View Forum Message <> Reply to Message

PS.: Kohait00 gets it right on spot too

Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by daveremba on Tue, 12 Jul 2011 22:15:54 GMT
View Forum Message <> Reply to Message

OK, your messages make it very clear & helpful.
No high level widgets.
I understand that in summary,
we're aiming for a low level port
at the System Draw level, etc.
(like from Windows to Linux).

Quote:Actually, I do not think so. There is only one possible path and it is the same as current
Linux and Windows backends...

Well within that path,
I am thinking of looking into the following
sub-paths:

1) X11 approach - develop in GCC
continue to use X11 code for drawing,
fix the observed artifacts (flat buttons and
uninitialized areas behind menus), and
add the needed code to get top windows recognized
by MacOS as true application processes
(most work appears to be done
since UPP already has X11/Linux done, but
X11 can be complex and emulator on
MacOS may not be 100% compliant)

2) OpenGL approach - develop in GCC
implement low level drawing code using OpenGL,
and use a simple library like GLUT for window
system, keybd, and mouse, events
(simple and high performance, but not sure
if GLUT behaves well with MacOS)
This is the approach used by:
http://www.openframeworks.cc/about

3) Objective-C++ approach - develop in Xcode
implement low level drawing, window, and
event code using Obj-C calls into MacOS (probably low level
Cocoa calls) hooked into SystemDraw
and other UPP libs.
(probably the most work, but maybe the
best visual result, and flexible towards
more compatbility with Apple in the future).

4) do whatever they do approach - develop in GCC
continue to look at other frameworks (wxWidgets, Firefox,
Qt, FLTK, Tcl/Tk) and see what they do;
borrow from their code libraries if possible.

wxWidgets uses approach (3) above

Here is an interesting comparison of
UI toolkits (from a wx perspective):
      http://wiki.wxwidgets.org/WxWidgets_Compared_To_Other_Toolki ts
wxWidgets has a pretty clear Cocoa library
in their source tree; unlike Firefox it
builds standalone apps rather hooked into
the Gecko framework (browser).

FLTK has an architecture that is most similar
to UPP - only a small portion of the code is
platform specific.  It draws its own
widgets using low level 2D drawing primitives
in OpenGL (as in approach (2) above).
Pros: it looks the same on all platforms,
Cons: it never looks like other native apps
on any platform.
This code may be helpful to fix problems
in approach (1 & 2) above.
Source code is here:
    http://www.fltk.org/software.php?VERSION=1.3.0&FILE=fltk /1.3.0/fltk-1.3.0-source.tar.gz

Firefox source is browsable here:
(to see how they do things, as you suggested Mirek)
http://mxr.mozilla.org/firefox/source/

The Firefox to Cocoa code is here:
http://mxr.mozilla.org/firefox/source/widget/src/cocoa/

I will do some tests/learning, and
report back in a few days, and
I will see how far the previous Mac
port effort got.

It seems that getting apps to work in the
X11 emulator is still worthwhile as the
easiest first step. (approach 1)

I spoke to my project client and they also do
want a working MacOS front end as well.

-Dave

Subject: Re: Porting (Mac OS X) and "reference application" idea

Posted by mirek on Wed, 13 Jul 2011 06:14:33 GMT

View Forum Message <> Reply to Message

daveremba wrote on Tue, 12 July 2011 18:15
3) Objective-C++ approach - develop in Xcode
implement low level drawing, window, and
event code using Obj-C calls into MacOS (probably low level
Cocoa calls) hooked into SystemDraw
and other UPP libs.
(probably the most work, but maybe the
best visual result, and flexible towards
more compatbility with Apple in the future).


Well, my intention actually was:

Objective-C/C++ approach - develop in TheIDE (X11).

So the steps are: Get TheIDE running in X11, investigate what is needed to be done to TheIDE build process to achieve _minimal_ cocoa application support, then use TheIDE to develop Cocoa based backend. Obiouslu, it is not important for macosx11 theide to look and feel native...

Of course, I can imagine that before we have Cocoa building X11 theide, somebody else can use xcode to start develop SystemDraw.

But I guess having theide is essential, as we need package/assembly system to do any reasonable development with U++ sources.

(That said, my immediate next plan after rainbow is "true" umake, one not requiring GUI to build apps, but that is another matter).

Subject: Re: Porting (Mac OS X) and "reference application" idea
Posted by kohait00 on Wed, 13 Jul 2011 08:06:26 GMT

View Forum Message <> Reply to Message

rainbow has a WinGL backend, which, as remember to have seen, also uses GLUT  maybe take a look in there too..