
Subject: Qt and Android...

Posted by [mirek](#) on Wed, 28 Apr 2010 16:32:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

Found this interesting thread:

http://groups.google.com/group/android-ndk/browse_thread/thread/a6c097fbd3acd22

If it is possible for Qt, it is possible for us as well

Mirek

Subject: Re: Qt and Android...

Posted by [dolik.rce](#) on Wed, 28 Apr 2010 17:52:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Interesting. If U++ gets supported on Android, I might start thinking about buying a new phone

BTW: The videos looked like there is some kind of emulator - that should make the porting much easier...

Honza

Subject: Re: Qt and Android...

Posted by [tojocky](#) on Wed, 28 Apr 2010 18:08:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Very Nice,

NDK is very powerful on Android!

luzr wrote on Wed, 28 April 2010 19:32 Found this interesting thread:

http://groups.google.com/group/android-ndk/browse_thread/thread/a6c097fbd3acd22

If it is possible for Qt, it is possible for us as well

Mirek

Subject: Re: Qt and Android...

Posted by [nneilson](#) on Wed, 28 Apr 2010 22:10:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

Android has capabilities that surprised me.
<http://www.youtube.com/watch?v=IICWWGQOYqc>
U++ should work OK.

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Wed, 05 May 2010 14:18:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

imagine developing U++ projects on Android, for remote controlling SQL and all the stuff... doing native C++ is preferable to java sometimes

thats even more interesting that porting to Apple Mac

Subject: Re: Qt and Android...
Posted by [nixnixnix](#) on Mon, 31 May 2010 01:13:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

If U++ can access the GPS through a virtual com port as well that would be fantastic! Has there been any progress on this? Is there likely to be a release of U++ for Android?

I've wanted to port my app to Android for a while but I wouldn't really know where to start. Java is not worth looking at for what I do as it is way too slow for a GIS style app.

Nick

Subject: Re: Qt and Android...
Posted by [cbpporter](#) on Mon, 31 May 2010 06:51:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

nixnixnix wrote on Mon, 31 May 2010 04:13: Is there likely to be a release of U++ for Android? That depends. Mirek, are you planning on buying a Nexus One anytime soon?

Subject: Re: Qt and Android...
Posted by [koldo](#) on Tue, 03 Aug 2010 07:00:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

- Android is now leading smartphones OS market
Android smart phone shipments grow 886% year-on-year in Q2 2010
<http://www.canalys.com/pr/2010/r2010081.html>

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Tue, 03 Aug 2010 09:11:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

we should not miss the train

Subject: Re: Qt and Android...
Posted by [koldo](#) on Tue, 03 Aug 2010 12:03:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

If somebody has some money, here there are some Android 7" tablets for 100\$, delivery included.

<http://www.dealextreme.com/search.dx/search.android>

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Tue, 03 Aug 2010 13:50:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

coool stuff...i might get me some

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Tue, 03 Aug 2010 14:19:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

unfortunately it's android 1.5, 1.6, 1.7.
current version is 2.1
didnt find out if its updateable..
but 100 bucks, cant do much wrong about it..

and discussions on the site reveal some problems to receive it after paying

Subject: Re: Qt and Android...
Posted by [manfhe](#) on Tue, 03 Aug 2010 17:49:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

To the beginning of the tests, can use the Android Emulator.

Android Emulator

Subject: Re: Qt and Android...
Posted by [koldo](#) on Tue, 03 Aug 2010 21:17:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oh, you see the name, "deal extreme"

A friend of mine told me about this address. The delivery lasted three weeks but he received the items, and they worked .

Coming to the porting, our disadvantage is that Nokia (Qt) is interested in mobile cross-platform developing.

However, our advantage is that porting Qt is much more difficult than porting U++.

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Wed, 04 Aug 2010 06:53:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

thats true,

nevertheless, i i'am lucky, i might finish porting U++ to framebuffer till the end of the month.. this should also work for android then, which uses framebuffer as underlying layer. but this would be a intermediate solution only. since android brings in own window manager which we should be using to make upp applications run just as the normal ones.

Subject: Re: Qt and Android...
Posted by [mirek](#) on Thu, 05 Aug 2010 10:45:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Wed, 04 August 2010 02:53thats true,

nevertheless, i i'am lucky, i might finish porting U++ to framebuffer till the end of the month.. this should also work for android then, which uses framebuffer as underlying layer. but this would be a intermediate solution only. since android brings in own window manager which we should be using to make upp applications run just as the normal ones.

BTW, possible interesting approach might be the "reversed library" (just made up the term).

Android certainly supports C libraries in Java. So make the U++ app as library and give it some Java encapsulation outer shell

Subject: Re: Qt and Android...
Posted by [mirek](#) on Fri, 06 Aug 2010 08:25:29 GMT

luzr wrote on Thu, 05 August 2010 06:45kohait00 wrote on Wed, 04 August 2010 02:53thats true, nevertheless, i i'am lucky, i might finish porting U++ to framebuffer till the end of the month.. this should also work for android then, which uses framebuffer as underlying layer. but this would be a intermediate solution only. since android brings in own window manager which we should be using to make upp applications run just as the normal ones.

BTW, possible interesting approach might be the "reversed library" (just made up the term).

Android certainly supports C libraries in Java. So make the U++ app as library and give it some Java encapsulation outer shell

Did some browsing:

<http://java.sun.com/docs/books/jni/html/fldmeth.html#11202>

Looks like calling java back from native code should be possible.

Given this fact, I wonder why is there so much fuss about Android not supporting native development. IMO, solution is simple - create Java interface class to Java SDK and pass a pointer to its instance to C++ code doing the actual stuff..

Subject: Re: Qt and Android...

Posted by [koldo](#) on Thu, 12 Aug 2010 11:48:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello all

Now the main mobile platforms are RIM (Blackberry), Android, Iphone, Meego (Maemo-Nokia+Moblin-Intel) and Windows Mobile/Phone.

The tools to develop in some of those platforms are:

- Android: Java, and partly C++(NDK)
- Meego: Qt C++, loosing support to Gtk+
- Windows Mobile: From version 7, only C#-Silverlight

So the situation is terrible for all...

I have seen some moves in forums:

- Qt is being slowly ported to Iphone and Android. No problem with MeeGo. No idea about mobile 7.
- WxWidgets is being ported to Qt-Maemo . They think Maemo will leave Gtk+. No Android and

mobile 7 move.

- CEGCC, the developers of a development environment for Windows CE/Mobile platforms are very upset about mobile 7. They are thinking to quit

Subject: Re: Qt and Android...

Posted by [koldo](#) on Thu, 12 Aug 2010 14:06:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

In summary, balancing technical issues and market forecasts, I would propose to invest our mobile devices time and efforts in Android.

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Thu, 12 Aug 2010 15:13:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

same opinion, iphone is popular but closed, Android grows faster..

Subject: Re: Qt and Android...

Posted by [Mindtraveller](#) on Thu, 12 Aug 2010 16:38:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

My vote for Android too.

iPhone could be the second choice.

Others is much less important for U++ future.

Subject: Re: Qt and Android...

Posted by [mirek](#) on Fri, 13 Aug 2010 06:57:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

OK. So it is agreed We are going Android.

Personally, I like the challenge - I think my solution to "NDK problem" should work.

Perhaps somebody could test it? (I mean, the possibility of calling Java code from NDK C++).

Subject: Re: Qt and Android...

Posted by [masu](#) on Fri, 13 Aug 2010 10:11:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have a Motorola Milestone with Android 2.1update1 installed, so I am able to test a few things ...

Matthias

Subject: Re: Qt and Android...

Posted by [jeremy_c](#) on Fri, 13 Aug 2010 10:41:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

This is great news and yes, Android is the best choice.

Jeremy

Subject: Re: Qt and Android...

Posted by [koldo](#) on Fri, 13 Aug 2010 12:57:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

Some thoughts

- As now Android has increasing momentum
 - And there are old established applications for PDA very difficult to port to Android
 - Perhaps now there is a gap for new applications made in, lets say, U++, to have a relevance in mobile devices
 - If we port U++ to Android not very late
-

Subject: Re: Qt and Android...

Posted by [mr_ped](#) on Fri, 13 Aug 2010 13:00:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hmm.. android is now hot stuff... and not only in good way.

Oracle sues Google for patent infringement

Subject: Re: Qt and Android...

Posted by [koldo](#) on Fri, 13 Aug 2010 17:50:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

mr_ped wrote on Fri, 13 August 2010 15:00Hmm.. android is now hot stuff... and not only in good way.

Oracle sues Google for patent infringement

Yes. I have the same opinion in other way. Without Java Android would be better

Quote:Oracle said in a statement that Google's Android system for mobile phones infringes on its patented Java technology.

Every successful device or project has to pass a lawsuit... I think Google will solve this.

Subject: Re: Qt and Android...

Posted by [koldo](#) on Sat, 14 Aug 2010 14:37:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

Excellent Miguel de Icaza post here

He ends it with:

Quote:Google could settle current damages with Oracle, and switch to the better designed, more pleasant to use, and more open .NET platform.

Miguel has been leader of Gnome project and Mono (C# open implementation).

I was disappointed when he began C#-Mono in Novell instead of doing a good C++ platform based on Gtk+ experience.

C++ imitators like Java or C# are filled with patent traps. Working close to Oracle (Java), C# (Microsoft) or Qt (Nokia) is not safe terrain.

Sorry Miguel, I would change your words to say:

Quote:Google could settle current damages with Oracle, and switch to the better designed, more pleasant to use, patent free and more open U++ platform.

Subject: Re: Qt and Android...

Posted by [Mindtraveller](#) on Sat, 14 Aug 2010 15:51:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

U++ is still a framework. It will be a platform after porting Framebuffer and Webkit. Still U++ has very strong requirements for developer level.

Subject: Re: Qt and Android...

Posted by [koldo](#) on Sat, 14 Aug 2010 19:16:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

And this is the list of OS identified in SourceForge:

Windows, Mac OS X, Linux, BSD, Solaris, and Android

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Wed, 18 Aug 2010 14:27:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

i am starting with setting up development environment eclipse for android. hello world is running in simulator already. have registered in android market as to be able to get a dev phone (nexus is out of stock though, till mid september).

now about NDK:

Quote:

Please note that the NDK does not enable you to develop native-only applications. Android's primary runtime remains the Dalvik virtual machine.

so the solution will be as Mirek showed, to setup a starter app for ultimate++ applications, which passes its 'surface' and 'input messages queue' to underlying ultimate code, using ndk maybe.

the picture still needs to be sharpened. i dont know how it could be possible to have access to all the java classes there are, i.e. for accessing gps and the like..will we have to produce wrapper classes for each and every thing we need 'down there in up'?

Subject: Re: Qt and Android...
Posted by [koldo](#) on Wed, 18 Aug 2010 15:22:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Good for your effort!

Yes Kohait:

Quote:the picture still needs to be sharpened.

There are ports to Android of Curl, SDL, Ffmpeg. We have to learn.

For example to read gps data could something like doing a

```
cat /dev/ttySO
```

(<http://stackoverflow.com/questions/2844384/how-to-define-gps-module-in-android>)

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Wed, 18 Aug 2010 20:07:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

definitely..

here is a opensource vnc client for android, maybe we can grab some ideas from it, how to get

and process events from user and how to draw things on android surface.

<http://code.google.com/p/android-vnc-viewer/>

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Thu, 19 Aug 2010 07:18:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

btw: how to deal with multitouch?

android (or actually the hardware mostly) supports multitouch, but U++ is plain old point and click.. would such a behaviour be serialized to multiple invokations? maybe android supports to report only first occurance (filtering only one)

Subject: Re: Qt and Android...

Posted by [koldo](#) on Thu, 19 Aug 2010 07:40:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Thu, 19 August 2010 09:18btw: how to deal with multitouch?

android (or actually the hardware mostly) supports multitouch, but U++ is plain old point and click.. would such a behaviour be serialized to multiple invokations? maybe android supports to report only first occurance (filtering only one)

Hello Kohait

For sure U++ interface will change slightly after porting to Android.

After analising Android SDK and NDK you can propose new methods for U++ main classes like Ctrl that match with multitouch or other features specific to new mobile devices.

It would be great to see soon in U++ code things like:

```
#ifdef ANDROID
```

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Thu, 19 Aug 2010 09:52:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

another thing i stumbled over is an NDK shipped example which upp could use to draw to..

Quote:

Android Bitmap objects from native code, and uses this to generate an old-school "plasma" effect.

altogether, it seems that mirek is right again, need to first provide a 'generic' portable interface for CtrlCore, which than can be enhanced to be a /dev/fb0 port or to be a android port.

in android case, there is surely a message notification mechanism, that intercepts key strokes etc..mouse clikcs..this 'simply' need to be translated to upp and forwarded 'down' to upp. the invokation every 10ms of the main thread procedure is to be ensured somehow though. no idea about that so far.

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Thu, 19 Aug 2010 12:40:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

i've successfully built the 'android-vnc-viewer' app from source, it runs on my emulated android. pretty impressive. now i'm trying to finish setup of my android sandbox making the NDK examples work. this is a bit hairy as it seems. as soon as this is running, i will provide a short descritioin on how to setup an android built environment, the information is available, but as always, the difficulties show up trying..especially NDK, which needs a cygwin for compiling the native source code into a library, which then will be linked when building the .apk android application itself.

another problem show up concerning the popup windows etc..they are TopWindow derived isn't it? so far, android app is an Activity which is merly a logic surface to place controls to and that can react on user interaction overriding several base class functions. so it's pretty much a Ctrl. but here, we wont be able to invoke popup windows that easy.. i think we will need the android means, if it has stuff like 'popup a surface and draw things on it while the rest is visible, blocking other stuff'.

what about the android look and feel? i am not that fit in creating the aproprate Ctrl.iml for android

Subject: Re: Qt and Android...

Posted by [koldo](#) on Thu, 19 Aug 2010 14:00:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Kohait

Perhaps the simplest focus to Android would be to begin with Core, leaving GUI issues a little bit later.

It would be encouraging to have a console "hello world" application compiled with TheIDE with some NTL code inside.

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Thu, 19 Aug 2010 15:25:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

this should be possible somehow, though printf redirection is not that easy especially to graphics...

an interesting topic is by the way the 'ContentProvider', means to storing and retrieving data from within application, basically data, that should survive onPause() time, when a user quits your application to switch somewhere else (as far as i understand)

i think it is closely related to their account management

Quote:

Content Providers

Content providers store and retrieve data and make it accessible to all applications. They're the only way to share data across applications; there's no common storage area that all Android packages can access.

Content providers are one of the primary building blocks of Android applications, providing content to applications. They encapsulate data and provide it to applications through the single ContentResolver interface. A content provider is only required if you need to share data between multiple applications. For example, the contacts data is used by multiple applications and must be stored in a content provider. If you don't need to share data amongst multiple applications you can use a database directly via SQLiteDatabase.

nice idea anyway, maybe upp could provide such an interface as well, this might reduce the need of Serialize persistence, but using SQL instead

Subject: Re: Qt and Android...

Posted by [koldo](#) on Thu, 19 Aug 2010 17:21:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Kohait

You are doing it very well.

However it is possible to do console programs .

This is an step by step sample to do a printf("Hello world"); program, including makefile and debugging.

<http://betelco.blogspot.com/2010/01/buildingdebugging-android-native-c.html>

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Thu, 19 Aug 2010 21:04:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

thanks, i'm just curious i'll try it ASAP.

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Fri, 20 Aug 2010 08:00:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

<http://code.google.com/p/android-pong/>

is another very simple free little app, probably more simple than a full blown vnc app.

i'll try to modify it soon, removing all unnessesary things and just leaving one paddle to show / grab user interaction and provide it here.

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Mon, 23 Aug 2010 13:29:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

here comes a pong version where the game logic has been striped. its pure startscreen, and moving padles. an example where quite a lot of underlying infrastructure is shown how to use it.

is anyone struggeling setting up eclipse btw? i've installed the galileo 3.5. version, adding it quite a bunch of plugins alltogether, and installing the current java sdk. this works pretty well from scratch, following the guide available at android.com. so this should be no problem. i thinkg ine needs to get familiar with this thing anyway, so i spare the work to describe how to set it all up, thus one digs in where it's needed at the source.

another question arised while i was reading code (i need to learn java anyway). the use things like

```
@Override
public void onDraw(Canvas canvas) {
    super.onDraw(canvas);
// draw operations on canvas
```

which is pretty much the idea of Draw interface and primitive calls on it. so the port to android should find a way to map the Upp calls DrawXYZ to respective calls to a canvas (a SystemDraw to a System canvas). this is not easy though, because of the java thing. but absolutely a must, because it uses hw accelerated stuff..

there are also native popups etc. how to do when porting the chameleon to android ? the common look and feel on android is sooo different..

File Attachments

1) [android-pong.rar](#), downloaded 408 times

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Tue, 24 Aug 2010 08:22:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

i've accomplished to intall cygwin and run an NDK based example (bitmap-plasma). so we could draw things .

on the JNI (java native interface), it is possible to call java stuff from c++ as well..

<http://java.sun.com/developer/onlineTraining/Programming/JDC Book/jni.html>

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Tue, 24 Aug 2010 09:11:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

a conceptional question:

android development is handled in eclipse, which is a good thing, for java, has android build stuff plugins etc..

the NDK based applications, are still java applications that use a libfoo.so packaged code to be linked in the .apk application file.

so this is the point that we need to produce. but we wont be able to provide a 'full' IDE support for java development (it's U++ after all..).

so the handling is to have a build method for this .so file, which is then coyied to a location in the android project, which is in last step built and pushed to the target using eclipse.

we neither will be able to debug it..thus the code / classes will need to be tested in Ultimate++ using test cases / test projects.

i'm trying to analyze the build process using the 'ndk-build' script, to be able to make own build method in Upp..the plasma example is my base.. maybe soon Core can be compiled..

EDIT: is it possible to define a buildmethod that acually uses the 'ndk-build' script? we would save ourselves a lot of hassle, needing only to define a Ultimate.mk file which declares the sources needed to be compiled also..

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Tue, 24 Aug 2010 11:13:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

this could impose some difficulties actually..

-fno-exceptions -fno-rtti

build flags from arm-eabi toolchain
how to deal with it?

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Tue, 24 Aug 2010 12:46:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

i found in the ide a 'SCRIPT' builder. is there some docu around it?
is it something i could use to map to ndk-builder?? providing some flags for it like execution dir
etc.?
any hints?

Subject: Re: Qt and Android...
Posted by [Sender Ghost](#) on Tue, 24 Aug 2010 20:00:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Tue, 24 August 2010 14:46i found in the ide a 'SCRIPT' builder. is there some
docu around it?

Hello, Konstantin.

You can read about SCRIPT builder from source code: `uppsrc/ide/Builders/ScriptBuilder.cpp`
Basicly, you have following built-in variables:

array INCLUDE
array LIBPATH

and built-in functions:

```
int Execute(string cmdLine)
void PutConsole(string text)
void PutVerbose(string text)
```

In the specified script file with bsc extension you need to implement (some of) following functions:

```
string objectfile(string pathToFile)
int compile(string pathToFile, string objectFile, string options)
string libraryfile(string packageName)
int library(array objlist, string product)
int link(objlist, liblist, target, linkoptions)
int preprocess(string file, string target)
```

kohait00 wrote on Tue, 24 August 2010 14:46any hints?

In my previous experience, ESC standard library need to be extended (e.g. with GetFileName, GetFileExt, etc. functions or they need to be implemented in the script file) and SCRIPT builder, as well (e.g. to get output directory).

Subject: Re: Qt and Android...
Posted by [kohait00](#) on Wed, 25 Aug 2010 06:38:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

thanks, sender ghost..
is there any bsc file i can look into? a builder script u use somewhere in the build / release process of upp?

Subject: Re: Qt and Android...
Posted by [Sender Ghost](#) on Wed, 25 Aug 2010 09:45:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Wed, 25 August 2010 08:38
is there any bsc file i can look into?

I made some template file for SCRIPT builder which you can look.
You can find it in attachment.

Edit:
Some mistakes fixed.

File Attachments

1) [BuilderScript.zip](#), downloaded 387 times

Subject: Re: Qt and Android...
Posted by [Sender Ghost](#) on Wed, 25 Aug 2010 09:58:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Also, you can be interested in lightweight ESC IDE.
It unfinished, but can be used as text editor with syntax highlighting based on uppsrc/CodeEditor.

With it you can test created functions inside main function using Print as output.

Edit:
Updated for newer changes in uppsrc/CodeEditor.

File Attachments

1) [EscIde_20191010.zip](#), downloaded 225 times

Subject: Re: Qt and Android...

Posted by [mirek](#) on Sun, 29 Aug 2010 19:49:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Tue, 24 August 2010 05:11a conceptional question:

android development is handled in eclipse, which is a good thing, for java, has android build stuff plugins etc..

the NDK based applications, are still java applications that use a libfoo.so packaged code to be linked in the .apk application file.

so this is the point that we need to produce. but we wont be able to provide a 'full' IDE support for java development (it's U++ after all..).

Actually, while little known, TheIDE already supports Java...

(Tom needed to develop something in Java).

Anyway, in this context, it is pershaps just interesting note, nothing more. You analysis seems correct..

Quote:

EDIT: is it possible to define a buildmethod that acually uses the 'ndk-build' script? we would save ourselves a lot of hassle, needing only to define a Ultimate.mk file which declares the sources needed to be compiled also..

Definitely. TheIDE should be modified to support Android development. Which perhaps invludes ndk-build. Maybe even some Java support.

Subject: Re: Qt and Android...

Posted by [kohait00](#) on Sun, 29 Aug 2010 20:12:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

i've managed to build an ndk example, but would definitely need some help with the build script 'ndk-build'. it includes actually those 2 steps only.

the ndk-build is invoked in the project directlry itself.

```
cd <your project main dir that contains also the jni folder>  
/<whatever path to ndk environmend>/ndk-build
```

if some help provided, i might handle to build a Core supprting first ndk based project.

another story is to definitely think about a better modularisation of code base. imagine every android app using upp taking up 3 MB ?? actually containing the same Core/Ctrl .so stuff..

Subject: Re: Qt and Android...

Posted by [mirek](#) on Mon, 30 Aug 2010 17:17:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

kohait00 wrote on Sun, 29 August 2010 16:12

another story is to definitely think about a better modularisation of code base. imagine every android app using upp taking up 3 MB ??

I see no problem with that. Cheapest android devices have >1GB of storage. Of course, RAM might be another question, but based on Linux, I guess process mapping will take care of that anyway.

And, practically, before we succeed, basic android devices will have more than 256MB.

I would not care about this now.
