Subject: QrCode image generation And debuging QR symbology
Posted by sergeynikitin on Sat, 31 Oct 2015 13:52:57 GMT
View Forum Message <> Reply to Message

New package to call QR code generator and make Image object to place to QTF and U++ applications

to compile under linux:

apt-get install libqrencode-dev

to compile under windows:
I can't to to test under windows.
It will be good to notice of details of success copmilation on windows systems.

File Attachments
1) ,
downloaded 581 times

Subject: Re: QrCode image generation And debuging QR symbology
Posted by sergeynikitin on Sat, 31 Oct 2015 13:54:32 GMT
View Forum Message <> Reply to Message

Package can help to debug QR-generation.

File Attachments
1) ,
downloaded 616 times

Subject: Re: QrCode image generation And debuging QR symbology
Posted by sergeynikitin on Sat, 31 Oct 2015 13:56:23 GMT
View Forum Message <> Reply to Message

Files of packages and test application.
1. QrEncode class

File Attachments
1) QrEncode.tar.gz, downloaded 266 times

Subject: Re: QrCode image generation And debuging QR symbology
Posted by sergeynikitin on Sat, 31 Oct 2015 13:58:09 GMT
View Forum Message <> Reply to Message

Files of packages and test application.
2. QrEncode test application

File Attachments
1) QrEncode_test.tar.gz, downloaded 258 times

Subject: Re: QrCode image generation And debuging QR symbology
Posted by sergeynikitin on Sat, 31 Oct 2015 13:59:49 GMT
View Forum Message <> Reply to Message

It will be pleaser if anybody review my code and point to codestyle mistakes.

Subject: Re: QrCode image generation And debuging QR symbology
Posted by Klugier on Sat, 31 Oct 2015 21:12:15 GMT
View Forum Message <> Reply to Message

Hello Sergey,

So you want to talk about your coding style? Ok let's start:

1. First of all. Never use "using namespace Upp" in your header file. Instead of this use NAMESPACE_UPP & END_UPP_NAMESPACE macro.

```
using namespace Upp;

class QrEncode {
// ...
};
```

Should be:

```
NAMESPACE_UPP

class QrEncode {
// ...
};

END_UPP_NAMESPACE
```

2. Next simply thing you do in your code regularly is (:

```
for(int i = 0; i < qrcode->width; i++){
```

Giv it a space before bracket:

```
for(int i = 0; i < qrcode->width; i++) {
```

3. The ternary operator (?:) should be written like this - more spaces:

```
int cs = casesensitive ? 1 : 0;
```

4. Decide where you put brackets in your if statement:

```
if ( qrcode == NULL ) {
```

It should be

```
if(qrcode == NULL) {
```

5. The valid way to represent if/else if/else block is:

```
if(...) {
   // ...
}
else
if(...) {
   // ...
}
else {
   // ...
}
```

You made small mistake here (never use this)

```
} else {
```

6. Method/Function arguments is not perfect:

```
is = Size(20+border*2,20+border*2);
```

It should be (a lot more spaces - remember to put space after comma, and after/before math operator like "-" or "+"):

```
is = Size(20 + border * 2, 20 + border * 2);
```

7. Operator "<<" should have spaces too:

```
uint8_t m = 1 << (layer);
```

8. Method/function opening braket starts always in new line:

```
Image QrEncode::QrEncode_To_Image(String s1){
```

Should be

```
Image QrEncode::QrEncode_To_Image(String s1)
{
```

9. You have got several tabulation/space problems - you can detect this by showing whitespace in your code or use feature show mismatch witespace in newer ide versions.

10. In class constructor QrEncode you have too many enters at the end.

P.S.
All things wrote in this post base on Ultimate++ code. Please notice that there is not official Ultimate++ coding standard. But I strongly believe that all things I proposed will increase your code read ability.

[EDIT - There is more!]
11. "public/private/protected" goes in the same line as class statment:

```
class QrEncoder {
  typedef QrEncode CLASSNAME;
  public:
    QrEncoder();
  // ...
```

It should be

```
class QrEncoder {
  typedef QrEncode CLASSNAME;
public:
  QrEncoder();
```

```
   // ...
```

Sincerely,
Klugier

Subject: Re: QrCode image generation And debuging QR symbology
Posted by sergeynikitin on Sat, 31 Oct 2015 21:28:23 GMT
View Forum Message <> Reply to Message

Thank you, Klugier!
But I use little different style.
By the way, I try to say some else thing:
interconnect between classes and functions.

Anyway, thank you.

Subject: Re: QrCode image generation And debuging QR symbology
Posted by Klugier on Sat, 31 Oct 2015 21:39:30 GMT
View Forum Message <> Reply to Message

Hello Sergey,

I understand that you can use your own coding standard in your private project. But if you want to post things in bazzar it should be a little bit more standardize. It is important, because users didn't want to read code that it is different than others. This is probably the reason why i don't like code in bazzar. The interesting thing here is that all core ultimate++ packages look very similar. Even when i contributed code to ultimate++ (look at the Android Builder code) I always wanted it to be compatibility with the rest of the code. And yes i used different coding standard for my "private" apps.

Please notice that some things in you code like "using namespace Upp" in header can lead to error in the future. It happens when user want to include your file, but didn't want to use namespace Upp.

You can also use const reference in String parameter:

```
Image QrEncode::QrEncode_To_Image(const String& s1)
```

Sincerely and thanks for discussion,
Klugier

Subject: Re: QrCode image generation And debuging QR symbology
Posted by mr_ped on Mon, 02 Nov 2015 18:07:17 GMT
View Forum Message <> Reply to Message

NAMESPACE_UPP doesn't open upp namespace for definitions? (too lazy to check the macro) So QrEncode would end up in Upp::QrEncode then? Which is probably not desired.

I would probably rather go for UPP::String instead of String ... (although now I wonder if it works when Upp namespace is switched off by some flag.. probably yes, as then upp classes end as global, and ::String is fine (notice I used UPP:: instead of Upp::, UPP should be well defined by U++ depending on that flag ... somewhere, not sure in which .h, but very likely some basic from Core)

Subject: Re: QrCode image generation And debuging QR symbology
Posted by Klugier on Mon, 02 Nov 2015 20:20:44 GMT
View Forum Message <> Reply to Message

Hello mr_ped,

You are right about "NAMESPACE_UPP", it equals "namespace Upp {". And yes in this case we have Upp::QrEncoder. Please notice that "using namespace Upp" force user to use this namespace in his/her file. So, this is definitely best alternative if you don't like to put "Upp::" before each things that refer to this namespace.

I don't know nothing about turning off namespace Upp. Is UPP not alias to Upp namespace. I mean "namespace UPP = Upp"?

Sincerely,
Klugier

Subject: Re: QrCode image generation And debuging QR symbology
Posted by mr_ped on Tue, 03 Nov 2015 01:31:59 GMT
View Forum Message <> Reply to Message

Normally UPP is Upp, so UPP::String is Upp::String.

But there is (or was) a way to compile upp without Upp namespace, at that point UPP is probably empty, so UPP::String is compiled as ::String I guess. Sorry, I'm too lazy to verify it.

Anyway, in Bazaar Packages, when you are creating .h file, I would go for UPP::upp_class way. Can't really see any downside about it, at least you see at first sight where the header is using the Upp classes, where it is using std::, and others. And the public header files should be as short and lean, as possible, so I will not accept "more writing" argument either.

Then the implementation in the .cpp is free to use "using namespace Upp;" to save typing, as the user of package usually doesn't need to read and understand implementation... as long, as the package works as the public header describes it. :d