
Subject: 2020.2rc1
Posted by [mirek](#) on Thu, 08 Oct 2020 10:33:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

First attempt at release:

<https://sourceforge.net/projects/upp/files/upp/2020.2rc1/>

Please test.

Subject: Re: 2020.2rc1
Posted by [Klugier](#) on Thu, 08 Oct 2020 10:46:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

I observed crash on mac (but only reproducible with the touchpad). All you need to do i just press command and execute scroll gesture on touchpad in code editor (TheIDE). The equivalent with mouse works correctly without crash.

Klugier

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Thu, 08 Oct 2020 17:24:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Klugier wrote on Thu, 08 October 2020 12:46Hello Mirek,

I observed crash on mac (but only reproducible with the touchpad). All you need to do i just press command and execute scroll gesture on touchpad in code editor (TheIDE). The equivalent with mouse works correctly without crash.

Klugier

I am unable to reproduce, there must be something special. If it is reproducible for you, can you e.g. post a screenshot just before it? Also, if it crashes in debug mode too, macos should provide you with stack trace, it would be very helpful to have it...

In fact, I think even release mode stack trace could provide me some pointers...

Mirek

Subject: Re: 2020.2rc1

Posted by [jimlef](#) on Thu, 08 Oct 2020 17:25:11 GMT

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

On Linux Mint 20, with my little program, no issues so far.

Subject: Re: 2020.2rc1

Posted by [Klugier](#) on Thu, 08 Oct 2020 20:15:58 GMT

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

Jim

This is macOS only issue and touchpad. On Linux and Windows everything works as expected.

I dig more with it and it seems like infinite loop not crash. On Linux and Windows everything works as expected. However, no luck with gdb with TheIDE on mac. However, with codesign I manage to launch executable (ide) inside gdb.

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Thu, 08 Oct 2020 21:47:46 GMT

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

Klugier wrote on Thu, 08 October 2020 22:15Jim

This is macOS only issue and touchpad. On Linux and Windows everything works as expected.

I dig more with it and it seems like infinite loop not crash. On Linux and Windows everything works as expected. However, no luck with gdb with TheIDE on mac. However, with codesign I manage to launch executable (ide) inside gdb.

Thanks!

Mirek

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Thu, 08 Oct 2020 21:48:15 GMT

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

jimlef wrote on Thu, 08 October 2020 19:25On Linux Mint 20, with my little program, no issues so far.

Thanks for reporting.

Mirek

Subject: Re: 2020.2rc1

Posted by [omari](#) on Fri, 09 Oct 2020 09:11:52 GMT

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

mirek wrote on Thu, 08 October 2020 18:24Klugier wrote on Thu, 08 October 2020 12:46Hello Mirek,

I observed crash on mac (but only reproducible with the touchpad). All you need to do i just press command and execute scroll gesture on touchpad in code editor (TheIDE). The equivalent with mouse works correctly without crash.

Klugier

I am unable to reproduce, there must be something special. If it is reproducible for you, can you e.g. post a screenshot just before it? Also, if it crashes in debug mode too, macos should provide you with stack trace, it would be very helpful to have it...

In fact, I think even release mode stack trace could provide me some pointers...

Mirek

I can reproduce it with this code:

```
#include <CodeEditor/CodeEditor.h>
```

```
GUI_APP_MAIN
```

```
{  
    Upp::CodeEditor ed;  
    Upp::TopWindow w;  
    w.SetRect(0, 0, 400, 400);  
    w.Add(ed.SizePos());  
    w.Run();  
}
```

just try to zoom in/out using two fingers while the command button is pressed.
here the stack trace:

File Attachments

1) [bt.txt.xz](#), downloaded 248 times

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Fri, 09 Oct 2020 15:17:08 GMT

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

Thank you, macos freeze issue should be now fixed. Please test.

Subject: Re: 2020.2rc1
Posted by [omari](#) on Fri, 09 Oct 2020 15:39:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, fixed.
Thank you Mirek.

Subject: Re: 2020.2rc1
Posted by [Oblivion](#) on Fri, 09 Oct 2020 18:11:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

EDIT: Nevermind. Not a bug. A local configuration problem...

Best regards,
Oblivion

Subject: Re: 2020.2rc1
Posted by [jimlef](#) on Sat, 10 Oct 2020 01:57:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Has anyone else tried to create a new project on mac? Can't seem to start in a valid folder...
Fresh unpacking of the package 15125 on Catalina.
Executing ~/upp/theide.app produces the following output when selecting new package:

Found the config file in ~/.config/u++/theide/ where MyApps.var contains

```
UPP =  
"/private/var/folders/fk/z0t14vn90gzb3gccg36z97lm0000gq/T/AppTranslocation/17F5F822-D129-4  
BEB-9E23-DEFA6BD71CCB/d/MyApps;/Users/jim/upp/uppsrc";  
OUTPUT = "/Users/jim/.cache/upp.out";
```

As I said, this was a first time install and run... no prior version.

File Attachments

1) [Untitled.png](#), downloaded 769 times

Subject: link errors with new(std::nothrow)
Posted by [wimpie](#) on Sun, 11 Oct 2020 15:00:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I was trying this release on windows and it gives link errors when using `new(std::nothrow)`

It is probably related to the issue discussed here

It's not a very big deal, I can change the original code and it also goes away when using `STD_NEWDELETE` flag but still wanted to mention it.

Anyway, keep up the great work! I'm still very happy with U++

regards,

Wimpie

Subject: Re: 2020.2rc1

Posted by [Klugier](#) on Sun, 11 Oct 2020 16:10:09 GMT

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

Hello,

Jim thanks for reporting. We will need to take a look at it before a release. We should have valid directory for MyApps after installation on macOS. This is important from the stability point of view.

I also find another bug. Whne executing following command line TheIDE opens Bombs, but when you open for example .iml file or one more time .cpp file the content of the file is empty (GNU/Linux):

```
./theide examples Bombs
```

The reason for that is that package files are open in uppsrc assembly instead of examples:

```
~/upp/uppsrc/Bombs/bombs.iml
```

When valid path is:

```
~/upp/examples/Bombs/bombs.iml
```

Klugier

Subject: Re: 2020.2rc1

Posted by [jimlef](#) on Sun, 11 Oct 2020 18:05:21 GMT

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

Klugier - I notice the same thing - when you click on say bombs.iml, it shows in the title line /home/james/MyApps/Bombs/Bombs.iml - notice there it's not the correct root directory.

Subject: Re: link errors with new(std::nothrow)
Posted by [mirek](#) on Sun, 11 Oct 2020 18:21:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

wimpie wrote on Sun, 11 October 2020 17:00Hello,

I was trying this release on windows and it gives link errors when using new(std::nothrow)

It is probably related to the issue discussed here

It's not a very big deal, I can change the original code and it also goes away when using STD_NEWDELETE flag but still wanted to mention it.

Anyway, keep up the great work! I'm still very happy with U++

regards,

Wimpie

Yes, I am afraid this will have to wait to be fixed in clang...

Mirek

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Sun, 11 Oct 2020 19:01:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Klugier wrote on Sun, 11 October 2020 18:10Hello,

Jim thanks for reporting. We will need to take a look at it before a release. We should have valid directory for MyApps after installation on macOS. This is important from the stability point of view.

I also find another bug. Whne executing following command line TheIDE opens Bombs, but when you open for example .iml file or one more time .cpp file the content of the file is empty (GNU/Linux):

./theide examples Bombs

The reason for that is that package files are open in uppsrc assembly instead of examples:

~/upp/uppsrc/Bombs/bombs.iml

When valid path is:

~/upp/examples/Bombs/bombs.iml

Klugier

Hopefully fixed.

Subject: Re: 2020.2rc1

Posted by [Klugier](#) on Sun, 11 Oct 2020 19:12:49 GMT

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

Hello Mirek,

Works like a charm :)

Klugier

Subject: Re: 2020.2rc1

Posted by [JeyCi](#) on Mon, 12 Oct 2020 04:49:31 GMT

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

mirek wrote on Thu, 08 October 2020 12:33First attempt at release.
is there a release for Windows-32x (with mingw)?

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Mon, 12 Oct 2020 07:49:42 GMT

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

JeyCi wrote on Mon, 12 October 2020 06:49mirek wrote on Thu, 08 October 2020 12:33First
attempt at release.

is there a release for Windows-32x (with mingw)?

No. I am sorry, but somehow I do not see the point having that...

Mirek

Subject: Re: 2020.2rc1

Posted by [JeyCi](#) on Mon, 12 Oct 2020 08:22:25 GMT

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

so, windows 32x is not supported anymore at all - in versions higher than 13664?.. or your answer concerns only v. with mingw compiler?.. (sorry that my question was so concrete about the compiler). and thanks for your reply

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Mon, 12 Oct 2020 09:49:51 GMT

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

JeyCi wrote on Mon, 12 October 2020 10:22so, windows 32x is not supported anymore at all - in versions higher than 13664?.. or your answer concerns only v. with mingw compiler?.. (sorry that my question was so concrete about the compiler). and thanks for your reply

You can produce windows 32-bit applications out of box (in Win64).

You just cannot run development environment on windows 32-bit. I believe that is sooo niche that it is not worth supporting... I do not see any reason why anybody would rather run 32-bit windows instead of 64 on development box.

BTW, chances that your CPU is not 64bit capable are virtually nil and you can upgrade to Win10 64bit for free...

Mirek

Subject: Re: 2020.2rc1

Posted by [Novo](#) on Mon, 12 Oct 2020 18:10:37 GMT

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

I tried to build Upp in Alpine Linux (the one used by Docker) and got the following error (in Debug):

```
/home/buildbot/dvlp/cpp/code/upp/git/uppsrc/Core/Util.cpp:49: undefined reference to `backtrace'  
/usr/bin/ld: /home/buildbot/dvlp/cpp/code/upp/git/uppsrc/Core/Util.cpp:50: undefined reference to  
`backtrace_symbols'
```

The problem can be fixed by adding of

```
DEBUG_LINK = "-lexecinfo";
```

to a build method.

But I believe this dependency should be detected automatically.

It would be great to fix that because that will make Upp easier to use with Docker.

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Mon, 12 Oct 2020 21:15:56 GMT

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

Novo wrote on Mon, 12 October 2020 20:10I tried to build Upp in Alpine Linux (the one used by Docker) and got the following error (in Debug):
/home/buildbot/dvlp/cpp/code/upp/git/uppsrc/Core/Util.cpp:49: undefined reference to `backtrace'
/usr/bin/ld: /home/buildbot/dvlp/cpp/code/upp/git/uppsrc/Core/Util.cpp:50: undefined reference to `backtrace_symbols'

The problem can be fixed by adding of

DEBUG_LINK = "-lexecinfo";

to a build method.

But I believe this dependency should be detected automatically.

It would be great to fix that because that will make Upp easier to use with Docker.

Or we can just remove backtrace? I do not remember a single time it helped and this is continuous nuisance on all systems...

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Mon, 12 Oct 2020 21:46:46 GMT

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

mirek wrote on Mon, 12 October 2020 23:15Novo wrote on Mon, 12 October 2020 20:10I tried to build Upp in Alpine Linux (the one used by Docker) and got the following error (in Debug):
/home/buildbot/dvlp/cpp/code/upp/git/uppsrc/Core/Util.cpp:49: undefined reference to `backtrace'
/usr/bin/ld: /home/buildbot/dvlp/cpp/code/upp/git/uppsrc/Core/Util.cpp:50: undefined reference to `backtrace_symbols'

The problem can be fixed by adding of

DEBUG_LINK = "-lexecinfo";

to a build method.

But I believe this dependency should be detected automatically.

It would be great to fix that because that will make Upp easier to use with Docker.

Or we can just remove backtrace? I do not remember a single time it helped and this is continuous nuisance on all systems...

I have put the whole function on #ifdef, so hopefully tomorrow release will be fixed...

Subject: Re: 2020.2rc1

Posted by [Novo](#) on Mon, 12 Oct 2020 22:21:59 GMT

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

mirek wrote on Mon, 12 October 2020 17:15

Or we can just remove backtrace? I do not remember a single time it helped and this is continuous nuisance on all systems...

I'm personally fine with that.

Your change has fixed the problem.
Thanks!

Ideally it would be great to provide Docker file/files for Upp. People seem to like Docker.

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Mon, 12 Oct 2020 22:46:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Tue, 13 October 2020 00:21mirek wrote on Mon, 12 October 2020 17:15
Or we can just remove backtrace? I do not remember a single time it helped and this is continuous
nuisance on all systems...
I'm personally fine with that.
Your change has fixed the problem.
Thanks!

Ideally it would be great to provide Docker file/files for Upp. People seem to like Docker.

I am in if you will provide me with some "seed" docker (so that I can automate it later). But not for
this release.

Mirek

Subject: Re: 2020.2rc1
Posted by [Novo](#) on Tue, 13 Oct 2020 00:22:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Mon, 12 October 2020 17:46
I have put the whole function on #ifdef, so hopefully tomorrow release will be fixed...
You also need to do the same thing to the code below
`#if defined(PLATFORM_POSIX) && defined(COMPILER_GCC) &&
!defined(PLATFORM_ANDROID)
include <execinfo.h>
include <cxxabi.h>
#endif`

execinfo.h is not available by default in Alpine Linux.

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Tue, 13 Oct 2020 07:55:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Tue, 13 October 2020 02:22mirek wrote on Mon, 12 October 2020 17:46

I have put the whole function on `#ifdef`, so hopefully tomorrow release will be fixed...
You also need to do the same thing to the code below
`#if defined(PLATFORM_POSIX) && defined(COMPILER_GCC) &&
!defined(PLATFORM_ANDROID)
include <execinfo.h>
include <cxxabi.h>
#endif`

`execinfo.h` is not available by default in Alpine Linux.

Done. Sorry for missing that, it was late...

Subject: Re: 2020.2rc1
Posted by [Klugier](#) on Tue, 13 Oct 2020 22:26:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Jim,

I manage to reproduce the issue with wrong "MyApps" folder on mac. What is bad is that our 2020.1 release is affected by this issue. We will work on the fix. One more time thanks for reporting!

When the new version of TheIDE will be available you will need to restart assemblies setup (to restore correct assembly path for MyApps). To do it just execute following line in your terminal:

```
rm -rf ~/.config/u++
```

Klugier

Subject: Re: 2020.2rc1
Posted by [jimlef](#) on Wed, 14 Oct 2020 02:02:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Glad I was able to give a little back :) Thanks for the tip!

Subject: Re: 2020.2rc1
Posted by [Klugier](#) on Wed, 14 Oct 2020 18:52:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Jim,

macOS issue should be solve toomorrow. Just clean cache as I suggest in previous post and

everything should work correctly. You could now port your new apps on macOS :)

I found one more issue. Not critical from the stability point of view, but for the general usability for users. A lot of uppsrc package has lack of description. When you add uppsrc dependency to your project you analyze this short description and on this basis you made the decision whenever package fits to your project/package or not.

Here is the screenshot:

Mirek, do we need package called plugin/zstd_legacy? Seems like something we do not want to have...

Klugier

File Attachments

1) [NoDescriptionsForUppsrcPackages.png](#), downloaded 651 times

Subject: Re: 2020.2rc1

Posted by [jimlef](#) on Wed, 14 Oct 2020 20:22:17 GMT

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

Thank you Klugier - will be on the lookout :)

Edit: Fresh install of U++ on Catalina (vm), installed command line tools, ran theide - and found out that .config/u++/theide/MyApps.var needed the MyApps path corrected.

Was set to "/Users/james/MyApps;/Users/james/upp/uppsrc"; out of the box... v. 15251.

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Thu, 15 Oct 2020 09:10:57 GMT

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

Klugier wrote on Wed, 14 October 2020 20:52

Mirek, do we need package called plugin/zstd_legacy? Seems like something we do not want to have...

Klugier

That depends on how much do you want to be U++ stable.

The problem here is that zstd changed the format since I have created the first plugin/zstd. Now if any user did use original plugin/zstd, he has created some files that cannot be uncompressed with the new plugin/zstd. That is the role of zstd_legacy, to support the original format.

Now I strongly suspect that I am the only zstd user and it is likely that removing _legacy would be fine, BUT in principle it is right to keep some form of backward compatibility at all times.

Mirek

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Thu, 15 Oct 2020 09:11:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Klugier wrote on Wed, 14 October 2020 20:52
I found one more issue. Not critical from the stability point of view, but for the general usability for users. A lot of uppsrc package has lack of description. When you add uppsrc dependency to your project you analyze this short description and on this basis you made the decision whenever package fits to your project/package or not.

I guess the description of most of these should be "Deprecated: do not use in the new code"...

Mirek

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Thu, 15 Oct 2020 13:30:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Package description & cleanup & README.md update done...

Subject: Re: 2020.2rc1
Posted by [Klugier](#) on Thu, 15 Oct 2020 16:19:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

Thanks for applying all these fixes and adding description. I think we are much more closer to 2020.2 release... I hope it will be the best U++ release ever.

Klugier

Subject: Re: 2020.2rc1
Posted by [Klugier](#) on Thu, 15 Oct 2020 20:18:19 GMT

Hello,

It seems that we have one more problem with template generation. By default in generating CtrlLib and Skylark application there is "using namespace Upp;" clausal in header file, which is wrong for obvious reasons.

Here is app generated with the template:

```
// ClockApp.h
#ifndef _Clock_Clock_h
#define _Clock_Clock_h

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

#define LAYOUTFILE <Clock/Clock.lay>
#include <CtrlCore/lay.h>

class ClockApp : public WithClockLayout<TopWindow> {
public:
    ClockApp();
};

#endif

// main.cpp
#include "Clock.h"

ClockApp::ClockApp()
{
    CtrlLayout(*this, "Window title");
}

GUI_APP_MAIN
{
    ClockApp().Run();
}
```

It should be:

```
// ClockApp.h
#ifndef _Clock_Clock_h
#define _Clock_Clock_h

#include <CtrlLib/CtrlLib.h>
```

```

namespace Upp {

#define LAYOUTFILE <Clock/Clock.lay>
#include <CtrlCore/lay.h>

class ClockApp : public WithClockLayout<TopWindow> {
public:
    ClockApp();
};

}

#endif

// main.cpp
#include "ClockApp.h"

using namespace Upp;

ClockApp::ClockApp()
{
    CtrlLayout(*this, "Window title");
}

GUI_APP_MAIN
{
    ClockApp().Run();
}

```

There is one option I would see there, but this is out of release scope. Upp namespace agnostic template (as option):

```

// ClockApp.h
#ifndef _Clock_Clock_h
#define _Clock_Clock_h

#include <CtrlLib/CtrlLib.h>

namespace Upp
{
    #define LAYOUTFILE <Clock/Clock.lay>
    #include <CtrlCore/lay.h>
}

class ClockApp : public Upp::WithClockLayout<Upp::TopWindow> {
public:

```

```

ClockApp();
};

#endif

// Clock.h
#include "ClockApp.h"

ClockApp::ClockApp()
{
    CtrlLayout(*this, "Window title");
}

GUI_APP_MAIN
{
    ClockApp().Run();
}

```

This option is the best for advanced users. In almost 90% your app shouldn't belong to Upp namespace. You could mix it easier with other namespaces like std. Only Core elements should belong there, but for simplicity we just put app code there...

In Turtle everything is fine. I think Oblivion fixed it some time ago.

Subject: Re: 2020.2rc1
 Posted by [mirek](#) on Thu, 15 Oct 2020 20:46:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Klugier wrote on Thu, 15 October 2020 22:18Hello,

It seems that we have one more problem with template generation. By default in generating CtrlLib and Skylark application there is "using namespace Upp;" clausal in header file, which is wrong for obvious reasons.

Which are? If we agreed with everything mainstream, there would be no U++ :)

Mirek

Subject: Re: 2020.2rc1
 Posted by [Klugier](#) on Thu, 15 Oct 2020 22:30:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

The idea is that someday we become mainstream :) Beside that we could introduce some

mainstream things still offering our uniqueness. In such case I do not think this big change in the philosophy or simplicity. I still think that we should keep with the best standards and practices for C/C++ if applicable. So moving in some areas towards mainstream solution is not something that we should be afraid of. This will give us broader horizons

Also, we have some mismatch between templates - TutrLte is using different namespace notation to that present in CtrlLib. We should unify this.

In the case of namespace agnostic we should give people choice (by simply checkbox in template). Anyway more and more people are using this approach. Example is available [here](#).

Anyway, the decision is always yours - you are the boss :)

Subject: Re: 2020.2rc1
Posted by [Oblivion](#) on Thu, 15 Oct 2020 22:41:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

There seems to be a problem with linking the latest freetype2 library (2.10.3) on Upp with GCC: 10.2.0, CLANG: 10.0.1, GNU ld: 2.35.

```
/usr/bin/ld: /home/maldoror/.cache/upp.out/Playground/Draw/GCC.Gui.Shared/Draw.a(FontFc.o):
in function `Upp::GetFontInfoSys(Upp::Font)':
FontFc.cpp:(.text._ZN3Upp14GetFontInfoSysENS_4FontE+0x11a): undefined reference to
`FT_Get_X11_Font_Format(FT_FaceRec_*)'
```

This can be fixed in FontFc.cpp, by putting the mentioned function into a C block:

```
extern "C" {
FT_EXPORT( const char* ) FT_Get_X11_Font_Format( FT_Face face ); // Put here to avoid
include path issues...
}
```

As a side note FT_Get_X11_Font_Format function name seems to be deprecated with v2.10.

Best regards,
Oblivion

Subject: Re: 2020.2rc1

Posted by [coolman](#) on Fri, 16 Oct 2020 07:40:28 GMT

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

Klugier wrote on Fri, 16 October 2020 00:30The idea is that someday we become mainstream :)
Beside that we could introduce some mainstream things still offering our uniqueness. In such case I do not think this big change in the philosophy or simplicity. I still think that we should keep with the best standards and practices for C/C++ if applicable. So moving in some areas towards mainstream solution is not something that we should afraid of. This will give us broader horizons

Also, we have some mismatch between templates - Tutrlte is using different namespace notaiton to that present in CtrlLib. We should unify this.

In the case of namespace agnostic we should give people choice (by simply checkbox in template). Anyway more and more people people are using this approach. Example is available here.

Anyway, the decision is always yours - you are the boss :)

Hi, I'm pleased with the example you have given. But at the same time, I must say that there is a small problem Assist++ and using namespace.

BR, Radek

Subject: Re: 2020.2rc1

Posted by [mirek](#) on Fri, 16 Oct 2020 07:55:04 GMT

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

Klugier wrote on Fri, 16 October 2020 00:30The idea is that someday we become mainstream :)
Beside that we could introduce some mainstream things still offering our uniqueness. In such case I do not think this big change in the philosophy or simplicity. I still think that we should keep with the best standards and practices for C/C++ if applicable. So moving in some areas towards mainstream solution is not something that we should afraid of. This will give us broader horizons

If we become mainstream, maybe so will become "using namespace" :)

Realistically, I agree with most things in "mainstream", but there are several dogmas that seem out of context. This is one of them...

Think about what you gain / loose by having / not having using namespace Upp in the header....

In practical terms, with using namespace, you can use U++ types without qualification at the risk of nameclash with either your own code or some other namespace that you have are "using". In the case of nameclash, you have to explicitly qualify namespace. Note that there also is no chance for error: nameclash is reported at compile time.

Without "using namespace", you have to qualify everything. Does not seem like a win option to me...

BTW, now thinking about it, I really dislike

```
namespace Upp {  
#define LAYOUTFILE <Rajce/Rajce.lay>  
#include <CtrlCore/lay.h>  
} // namespace Upp
```

- that is something we should fix in the next release. This should work without Upp... I think the most correct fix is in the Layout designer - add namespace setting to .usc file and make layout designer to use it / insert into .lay....

Mirek

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Fri, 16 Oct 2020 09:26:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oblivion wrote on Fri, 16 October 2020 00:41Hi,

There seems to be a problem with linking the latest freetype2 library (2.10.3) on Upp with GCC: 10.2.0, CLANG: 10.0.1, GNU ld: 2.35.

```
/usr/bin/ld: /home/maldoror/.cache/upp.out/Playground/Draw/GCC.Gui.Shared/Draw.a(FontFc.o):  
in function `Upp::GetFontInfoSys(Upp::Font)':  
FontFc.cpp:(.text._ZN3Upp14GetFontInfoSysENS_4FontE+0x11a): undefined reference to  
`FT_Get_X11_Font_Format(FT_FaceRec_*)'
```

This can be fixed in FontFc.cpp, by putting the mentioned function into a C block:

```
extern "C" {  
FT_EXPORT( const char* ) FT_Get_X11_Font_Format( FT_Face face ); // Put here to avoid  
include path issues...  
}
```

As a side note FT_Get_X11_Font_Format function name seems to be deprecated with v2.10.

Best regards,

Oblivion

FT_Get_X11_Font_Format now removed: the info that it provides is not much relevant anyway (we used it to check that font is truetype - scaleable, but today all fonts are and U++ unconditionally expects that anyway) and I have replaced the test with testing the font path extension.

I hope this was the last one for 2020.2 :)

Mirek

Subject: Re: 2020.2rc1
Posted by [Klugier](#) on Fri, 16 Oct 2020 09:49:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

I just noticed on macOS (latest nightly and macOS 10.15 catalina) that when option from menubar open new window the highlight in global menu stays. Screenshot below:

I remembered we had similar issue in the past and it seems like the regression...

Klugier

File Attachments

1) [Screenshot 2020-10-16 at 11.46.58.png](#), downloaded 540 times

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Fri, 16 Oct 2020 12:13:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Klugier wrote on Fri, 16 October 2020 11:49Hello Mirek,

I just noticed on macOS

Just stop it :)

OK, I am going to consider this more of cosmetics issue. I will file it in RM; if there are no more issues, this is not blocker, if there are, I will fix it in next rc3...

Mirek

Subject: Re: 2020.2rc1
Posted by [mirek](#) on Tue, 20 Oct 2020 07:40:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

In case it was missed, there is now rc2 on the sf.net....

As there no complaints so far, I am thinking about releasing soon...

Subject: Re: 2020.2rc1
Posted by [Tom1](#) on Tue, 20 Oct 2020 19:03:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mirek,

Not missed! I have been working with this since day one it was released and have no complaints so far.

Thanks and best regards,

Tom

EDIT: Well not this, but the RC2 of course!
