

---

Subject: New release 2024 "alpha phase"  
Posted by [mirek](#) on Sun, 08 Sep 2024 19:21:24 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

2024 (rev. >17373) (Sep 2024)

Upcoming release

Core

First release of U++ that requires C++17

Moveable concept redesigned with C++17 features. U++ now allows non-moveable types to be stored in Vector flavor of containers (using `Upp::is_upp_guest`). PODs are automatically moveable (aka trivially relocatable)

`Upp::Tuple` now supports structured binding

`GetFileTime`, `GetFileLength`, `FileExists`, `DirectoryExists` and `FileMapping` refactored

`Stream::GetAll` now invokes `LoadError` on negative count

`ValueCache` limits setting methods are simplified

`Value` now directly supports 'float' type

Some iffy code now made more C++ compliant (e.g. always using `memcpy` for unaligned data)

`AsXML` had new `XML_ESCAPELF`

plugin/Zip

zip64 support

Draw

UHD image now can serve as source for SD image

New `S3` .iml image flag - the images are drawn supersampled 3x, usually without antialiasing, and only downsampled at runtime

Painter

Multithreaded rendering further optimised

New image filtering parameter - so far, rendering image was always with bilinear filtering, new parameter allows other Image filter like Lanczos 3

CtrlCore

Horizontal mouse scroll wheel support

`CtrlMapper` now provides `operator()(Ctrl, T, const T& factor)` for simple unit conversions

gtk backend improvements, XWayland mouse cursor bug workaround

CtrlLib

`CtrlMapper` now provides `operator()(Ctrl, T, const T& factor)` for simple unit conversions

ide

Icon Designer refactored and optimised, new tools added, S3 flag support added  
Alt-M now goes to special scratchpad file of the same type as is current file, this is helpful e.g. for temporary storing and editing parts of .iml images that are then composed to the final image.  
Output directory in assembly definition now can be left empty and defaults to reasonable path.  
Hexadecimal view is now much faster  
Fixed further corner case Assist++ problems  
Layout designer text field, used with e.g. Labels, now has Qtf button to edit text with RichEdit  
Git file history now goes through renames  
Compare with menu now suggests files in Download folder too  
Main package configuration dialog improved

plugin upgrades

plugin/sqlite3: 3.46.0  
plugin/lzma: 24.6  
plugin/zstd: 1.5.6  
Core: LZ4 1.9.4  
plugin/z: 1.3.1  
plugin/png: 1.6.46  
plugin/tif: 4.6.0  
plugin/jpeg: 9f

Win32

OpenSSL upgraded to 3.2.1  
Clang compiler upgraded to 18.1.5

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Sun, 08 Sep 2024 19:23:34 GMT  
[View Forum Message](#) <> [Reply to Message](#)

This is long overdue, I was a bit too busy in the first half of year...

Please use this thread to alert me of anything serious that should eventually be addressed before the release!

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Klugier](#) on Sun, 08 Sep 2024 21:22:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

It's good to hear that the new release is on the radar. For the C++17 I repeated it many times that we should adopt it. Probably starting in 2020, but finally we have it

Threads with C++17:

- [https://www.ultimatepp.org/forums/index.php?t=msg&th=111\\_41&goto=54751&](https://www.ultimatepp.org/forums/index.php?t=msg&th=111_41&goto=54751&)
- [https://www.ultimatepp.org/forums/index.php?t=msg&th=117\\_72&goto=58266&#msg\\_58266](https://www.ultimatepp.org/forums/index.php?t=msg&th=117_72&goto=58266&#msg_58266)

Klugier

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Sun, 08 Sep 2024 21:38:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Klugier wrote on Sun, 08 September 2024 23:22Hello Mirek,

It's good to hear that the new release is on the radar. For the C++17 I repeated it many times that we should adopt it. Probably starting in 2020, but finally we have it

Threads with C++17:

- [https://www.ultimatepp.org/forums/index.php?t=msg&th=111\\_41&goto=54751&](https://www.ultimatepp.org/forums/index.php?t=msg&th=111_41&goto=54751&)
- [https://www.ultimatepp.org/forums/index.php?t=msg&th=117\\_72&goto=58266&#msg\\_58266](https://www.ultimatepp.org/forums/index.php?t=msg&th=117_72&goto=58266&#msg_58266)

Klugier

But there is a logic to it. It is complicated to produce working binary of your application on 10 years old distro if it uses c++ 17, which unfortunately, believe or not, some consumers require... What C++17 adds is good, but does not fundamentally change the way we are using C++ (unlike C++11/14). Makes sense to lag a bit.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [dodobar](#) on Sun, 08 Sep 2024 22:30:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Awesome, thanks for putting the time into another release.

looks like Christmas is coming early...

Curt

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Mon, 09 Sep 2024 05:59:59 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Mirek,

Looking forward to the release...

Meanwhile, can you take a look at the MSC compilation (MSBT22x64 in my case):  
C:\upp-git\upp.src\uppsrc\Core\Topt.h(168): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Topt.h(172): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Tuple.h(170): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Tuple.h(174): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Tuple.h(179): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Complex.h(58): error C7525: inline variables require at least '/std:c++17'

EDIT: Adding '/std:c++17' to Common options in build methods for MSC helped here.

Best regards,

Tom

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Mon, 09 Sep 2024 07:50:42 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Mon, 09 September 2024 07:59Hi Mirek,

Looking forward to the release...

Meanwhile, can you take a look at the MSC compilation (MSBT22x64 in my case):  
C:\upp-git\upp.src\uppsrc\Core\Topt.h(168): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Topt.h(172): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Tuple.h(170): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Tuple.h(174): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Tuple.h(179): error C7525: inline variables require at least '/std:c++17'  
C:\upp-git\upp.src\uppsrc\Core\Complex.h(58): error C7525: inline variables require at least '/std:c++17'

EDIT: Adding '/std:c++17' to Common options in build methods for MSC helped here.

Best regards,

Tom

Added to instant setup, existing installation has to be fixed by adding it to build method C++ options.

Mirek

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Mon, 09 Sep 2024 08:54:28 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thanks Mirek,

Works fine now. I'll keep on testing.

Best regards,

Tom

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Mon, 09 Sep 2024 13:07:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

A compiler warning popped up on 32-bit MSBT22:

C:\upp-git\upp.src\uppsrc\Core\FileMapping.h (18): warning C4244: 'argument': conversion from 'Upp::int64' to 'size\_t', possible loss of data

BTW: which condition flags should I use to include specific libraries only for 64-bit targets in TheIDE Package organizer? E.g. WIN64 or \_WIN64 does not seem to work here.

Best regards,

Tom

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Tue, 10 Sep 2024 12:40:18 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Mon, 09 September 2024 15:07A compiler warning popped up on 32-bit MSBT22:

```
C:\upp-git\upp.src\uppsrc\Core\FileMapping.h (18): warning C4244: 'argument': conversion from 'Upp::int64' to 'size_t', possible loss of data
```

Thanks, fixed.

Quote:

BTW: which condition flags should I use to include specific libraries only for 64-bit targets in TheIDE Package organizer? E.g. WIN64 or \_WIN64 does not seem to work here.

Best regards,

Tom

That is suprisingly complicated problem... Build method actually does not really know / care about that. And frankly, so far, there were no specific library names for 64-bit, usually in win32, they are just placed to different folders.

If you think this is really worth fixing, the simple way is to put something into Common fixed flags in the build method. I can do that in automatic build method setup... I guess placing "M32" to 32-bit build methods should do the job.

Mirek

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Tue, 10 Sep 2024 15:15:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Tue, 10 September 2024 15:40Tom1 wrote on Mon, 09 September 2024 15:07A compiler warning popped up on 32-bit MSBT22:

```
C:\upp-git\upp.src\uppsrc\Core\FileMapping.h (18): warning C4244: 'argument': conversion from 'Upp::int64' to 'size_t', possible loss of data
```

Thanks, fixed.

Quote:

BTW: which condition flags should I use to include specific libraries only for 64-bit targets in TheIDE Package organizer? E.g. WIN64 or \_WIN64 does not seem to work here.

Best regards,

Tom

That is suprisingly complicated problem... Build method actually does not really know / care about that. And frankly, so far, there were no specific library names for 64-bit, usually in win32, they are just placed to different folders.

If you think this is really worth fixing, the simple way is to put something into Common fixed flags in the build method. I can do that in automatic build method setup... I guess placing "M32" to 32-bit build methods should do the job.

Mirek  
Hi,

Some external libraries (and DLLs) I use come with just different names for 32/64 bit variants. It would be helpful to have a way to use e.g. "WIN32 M32" or "WIN32 !M32" to link with the correct variant and avoid linking errors.

Best regards,

Tom

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Wed, 11 Sep 2024 08:48:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Tue, 10 September 2024 18:15mirek wrote on Tue, 10 September 2024 15:40Tom1 wrote on Mon, 09 September 2024 15:07A compiler warning popped up on 32-bit MSBT22:

```
C:\upp-git\upp.src\uppsrc\Core\FileMapping.h (18): warning C4244: 'argument': conversion from 'Upp::int64' to 'size_t', possible loss of data
```

Thanks, fixed.

Quote:

BTW: which condition flags should I use to include specific libraries only for 64-bit targets in TheIDE Package organizer? E.g. WIN64 or \_WIN64 does not seem to work here.

Best regards,

Tom

That is suprisingly complicated problem... Build method actually does not really know / care about that. And frankly, so far, there were no specific library names for 64-bit, usually in win32, they are just placed to different folders.

If you think this is really worth fixing, the simple way is to put something into Common fixed flags in the build method. I can do that in automatic build method setup... I guess placing "M32" to

32-bit build methods should do the job.

Mirek  
Hi,

Some external libraries (and DLLs) I use come with just different names for 32/64 bit variants. It would be helpful to have a way to use e.g. "WIN32 M32" or "WIN32 !M32" to link with the correct variant and avoid linking errors.

Best regards,

Tom

Thanks for the advise Mirek. Adding a 'M32' in 'Common fixed flags' for 32-bit CLANG allowed using condition 'WIN32 !M32' for a 64-bit DLL to prevent the linking error. It probably would not hurt to have a M32 and/or M64 flags pre-defined for the corresponding builders to make this work out of the box.

Best regards,

Tom

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Wed, 11 Sep 2024 09:36:48 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Mirek,

I started to have trouble seeing some of the TheIDE packages' names properly. Then I noticed that the Dark theme coloring has changed.

Especially Draw, PdfDraw and all <aux> packages are really hard to read for me. The screenshot is from Linux Mint using dark theme, but I'm having the same issue on Windows dark theme too, although the darker background helps there a little bit.

(I must remind you that I have red-green colorblindness, like 8 % of European men do. Not sure if this contributes to the issue I'm having now with those texts. To simulate red-green colorblindness to the people with normal color perception, try to flatten your red color channel to something like 0.. 10 % of the original and see how your GUI looks. Might be a good idea to do this test with any GUI design to ensure accessibility for guys like me.)

Best regards,

Tom

EDIT: Sorry, I seem to have misplaced this posting... Just found there's a thread for this at [https://www.ultimatepp.org/forums/index.php?t=msg&th=12329&goto=60794&#msg\\_60794](https://www.ultimatepp.org/forums/index.php?t=msg&th=12329&goto=60794&#msg_60794) , so feel free to move this posting to that thread.

## File Attachments

---

1) [Screenshot from 2024-09-11 12-07-32.png](#), downloaded 553 times

File Edit Project Build Debug Assist Setup Help



UWord

Core

CtrlCore

CtrlLib

Draw

Painter

PdfDraw

RichEdit

RichText

plugin/bmp

plugin/jpg

plugin/png

plugin/z

&lt;prj-aux&gt;

&lt;ide-aux&gt;

&lt;temp-aux&gt;

&lt;meta&gt;

UWord.cpp

UWord.iml

UWord.cpp x UWord.cpp x UW

*Symbol/lineno (Ctrl+G)*

\*

UWord

UWord/UWord.cpp

UWord/UWord.cpp

IMAGECLASS

IMAGEFILE

UWordFs() → FileSel&amp;

PdfFs() → FileSel&amp;

UWord

UWord : public TopWindow

DragAndDrop(Point, PasteClip&amp; d

FrameDragAndDrop(Point, PasteC

editor → RichEdit

menubar → MenuBar

toolbar → ToolBar

statusbar → StatusBar

filename → String

lrufile() → static LRUList&amp;

Load(const String&amp; filename) → v

OpenFile(const String&amp; fn) → void

New() → void

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Thu, 12 Sep 2024 08:57:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Wed, 11 September 2024 11:36Mirek,

I started to have trouble seeing some of the TheIDE packages' names properly. Then I noticed that the Dark theme coloring has changed.

Especially Draw, PdfDraw and all <aux> packages are really hard to read for me. The screenshot is from Linux Mint using dark theme, but I'm having the same issue on Windows dark theme too, although the darker background helps there a little bit.

(I must remind you that I have red-green colorblindness, like 8 % of European men do. Not sure if this contributes to the issue I'm having now with those texts. To simulate red-green colorblindness to the people with normal color perception, try to flatten your red color channel to something like 0.. 10 % of the original and see how your GUI looks. Might be a good idea to do this test with any GUI design to ensure accessibility for guys like me.)

Best regards,

Tom

EDIT: Sorry, I seem to have misplaced this posting... Just found there's a thread for this at [https://www.ultimatepp.org/forums/index.php?t=msg&th=12329&goto=60794&#msg\\_60794](https://www.ultimatepp.org/forums/index.php?t=msg&th=12329&goto=60794&#msg_60794) , so feel free to move this posting to that thread.

For now, the changed to DarkTheme are rolled back. It will need some more research I guess...

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Thu, 12 Sep 2024 08:59:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quote:

It probably would not hurt to have a M32 and/or M64 flags pre-defined for the corresponding builders to make this work out of the box.

Done, but in the end I decided to change it to CPU\_32.

Mirek

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Thu, 12 Sep 2024 09:28:05 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Thu, 12 September 2024 11:57...

For now, the changed to DarkTheme are rolled back. It will need some more research I guess... Still cannot see them well. I looked at the old (2020) code and the values (in old coding style) were:

```
double DarkTheme_c[3] = { 0.3, 0.5, 0.2 };  
int   DarkTheme_middle = 155;
```

Those I can see well on the dark background.

Best regards,

Tom

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Thu, 12 Sep 2024 09:31:16 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Ah, sorry, proper rollback pushed.

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Thu, 12 Sep 2024 09:45:29 GMT  
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Thu, 12 September 2024 11:59Quote:  
It probably would not hurt to have a M32 and/or M64 flags pre-defined for the corresponding builders to make this work out of the box.

Done, but in the end I decided to change it to CPU\_32.

Mirek

Thanks! Works fine, although having it change the output directory to include CPU32 is not needed nor desired.

// Tom

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Thu, 12 Sep 2024 09:48:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Another issue... On Linux Mint and UHD 4k display the console output (when compiling) is using a very tiny little font... And even if I zoom it in using Ctrl+Wheel, on next startup of TheIDE it has shrunk again back. All other panels in TheIDE seem to follow the UHD sizing.

// Tom

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [omari](#) on Thu, 12 Sep 2024 11:53:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Nightly build is stopped.

it seems that the builder host need update in order to use C++17

Thanks.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Fri, 13 Sep 2024 17:42:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Thu, 12 September 2024 11:48Another issue... On Linux Mint and UHD 4k display the console output (when compiling) is using a very tiny little font... And even if I zoom it in using Ctrl+Wheel, on next startup of TheIDE it has shrunked again back. All other panels in TheIDE seem to follow the UHD sizing.

// Tom

What is in Setup/Settings? (Maybe screenshot...)

Also, TheIDE font sizes are set once on the very first start. Perhaps you started in HD and only later switched to UHD ?

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Fri, 13 Sep 2024 18:54:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Fri, 13 September 2024 20:42Tom1 wrote on Thu, 12 September 2024 11:48Another issue... On Linux Mint and UHD 4k display the console output (when compiling) is using a very tiny little font... And even if I zoom it in using Ctrl+Wheel, on next startup of TheIDE it has shrunked again back. All other panels in TheIDE seem to follow the UHD sizing.

// Tom

What is in Setup/Settings? (Maybe screenshot...)

Also, TheIDE font sizes are set once on the very first start. Perhaps you started in HD and only later switched to UHD ?

---

It's entirely possible that it was mixed up on the very first start. Cannot check right now as the issue surfaced at the office. My home installation seems to work just fine. Also, I did not know that changing console font by Ctrl+Wheel is not persistent. (In the code editor it seems to be.)

I guess this can be considered solved. I will let you know next week if this is not the case once I check my office setup.

Thanks and best regards,

Tom

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [omari](#) on Sat, 14 Sep 2024 10:11:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

## File Attachments

1) [\\_upp-nightly.PNG](#), downloaded 477 times

- Status & Roadmap
- FAQ
- Authors & License
- Forums
- Funding U++

Search on this site

 Site search

 Like 78



## Nightly builds

Date	U++ for Windows (with CLANG)	U++ for Linux/MacOS/BSD/Solaris
2024-09-14		
2024-09-13		
2024-09-12		
2024-09-11		
2024-09-10		
2024-09-09		
2024-09-08		
2024-09-07		
2024-09-06		
2024-09-05		
2024-09-04	01:07 <a href="#">upp-win-17348.7z</a> (156 MB)	01:07 <a href="#">upp-posix-17348.tar.xz</a> (36 MB)
2024-09-03		
2024-09-02		
2024-09-01		

---

Subject: Re: New release 2024 "alpha phase"

Posted by [mirek](#) on Sun, 15 Sep 2024 13:41:10 GMT

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

---

omari wrote on Thu, 12 September 2024 13:53Nightly build is stopped.

it seems that the builder host need update in order to use C++17

Thanks.

Should be now fixed.

Mirek

---

---

Subject: Re: New release 2024 "alpha phase"

Posted by [Tom1](#) on Mon, 16 Sep 2024 08:21:05 GMT

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

---

Tom1 wrote on Fri, 13 September 2024 21:54mirek wrote on Fri, 13 September 2024 20:42Tom1 wrote on Thu, 12 September 2024 11:48Another issue... On Linux Mint and UHD 4k display the console output (when compiling) is using a very tiny little font... And even if I zoom it in using Ctrl+Wheel, on next startup of TheIDE it has shrunked again back. All other panels in TheIDE seem to follow the UHD sizing.

// Tom

What is in Setup/Settings? (Maybe screenshot...)

Also, TheIDE font sizes are set once on the very first start. Perhaps you started in HD and only later switched to UHD ?

It's entirely possible that it was mixed up on the very first start. Cannot check right now as the issue surfaced at the office. My home installation seems to work just fine. Also, I did not know that changing console font by Ctrl+Wheel is not persistent. (In the code editor it seems to be.)

I guess this can be considered solved. I will let you know next week if this is not the case once I check my office setup.

Thanks and best regards,

Tom

Hi,

Settings were showing small font size for all but the 'Normal', which I had probably changed by Ctrl+Wheel. So I must have initially started in HD mode.

First I tried to fix it by using 'Restore defaults' and then 'Close'. This caused TheIDE to crash on every attempt. Then I selected the same font sizes manually and the values were successfully updated after close. So, there must be some issue with 'Restore defaults'.

Anyway, now I have the correct UHD font sizes.

Thanks and best regards,

Tom

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Tue, 17 Sep 2024 07:10:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

[quote title=Tom1 wrote on Mon, 16 September 2024 10:21][quote title=Tom1 wrote on Fri, 13 September 2024 21:54]mirek wrote on Fri, 13 September 2024 20:42First I tried to fix it by using 'Restore defaults' and then 'Close'. This caused TheIDE to crash on every attempt. Then I selected the same font sizes manually and the values were successfully updated after close. So, there must be some issue with 'Restore defaults'.

Tom

Should be now fixed.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Tom1](#) on Tue, 17 Sep 2024 08:43:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Tue, 17 September 2024 10:10Tom1 wrote on Mon, 16 September 2024 10:21First I tried to fix it by using 'Restore defaults' and then 'Close'. This caused TheIDE to crash on every attempt. Then I selected the same font sizes manually and the values were successfully updated after close. So, there must be some issue with 'Restore defaults'.

Tom

Should be now fixed.

Thanks!

// Tom

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Tue, 17 Sep 2024 15:34:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thank you mirek for all the hard work!

I have a question. In CoWork.h, line 186, etc,

```
void Do(Function&& f, Args&&... args) { co.Do([=]() { ret = f(args...); }); }
```

shouldn't it be something like

```
void Do(Function&& f, Args&&... args) { co.Do([=]() { ret = f(std::forward<Args>(args)...); }); }
```

for perfect forwarding to actually work?

The current upp codebase works fine with std=c++20, except the warning: implicit capture of 'this' with a capture default of '=' is deprecated [-Wdeprecated-this-capture] issue. This can be fixed without much effort and without affecting the aimed standard of c++17. I was wondering if you would consider accepting it if I make a pull request for that.

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Wed, 18 Sep 2024 09:44:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Lance wrote on Tue, 17 September 2024 17:34 Thank you mirek for all the hard work!

I have a question. In CoWork.h, line 186, etc,

```
void Do(Function&& f, Args&&... args) { co.Do([=]() { ret = f(args...); }); }
```

shouldn't it be something like

```
void Do(Function&& f, Args&&... args) { co.Do([=]() { ret = f(std::forward<Args>(args)...); }); }
```

for perfect forwarding to actually work?

Interestingly, does not work.

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

```
using namespace Upp;
```

```
CONSOLE_APP_MAIN
```

```
{
auto a = Async([](int n) -> double {
double f = 1;
for(int i = 2; i <= n; i++)
f *= i;
return f;
}, 100);
}
```

is compile time error with `std_forward`. Not sure why, but for now have to revert the change.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Wed, 18 Sep 2024 09:49:51 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Lance wrote on Tue, 17 September 2024 17:34

The current upp codebase works fine with `std=c++20`, except the warning: implicit capture of 'this' with a capture default of '=' is deprecated [-Wdeprecated-this-capture] issue. This can be fixed without much effort and without affecting the aimed standard of `c++17`. I was wondering if you would consider accepting it if I make a pull request for that.

Nope, cannot do that with `C++17`. Requires `[=, this]`, which is `C++20` feature. And, frankly, while `[=]` capturing 'this' is perhaps confusing, everybody is used to it by now so to require longer code for the same thing is at this point is kind of pointless... But when we go `C++20`, we shall do it anyway. In 3 years...

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Wed, 18 Sep 2024 13:21:28 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Lance wrote on Tue, 17 September 2024 17:34 Thank you mirek for all the hard work!

I have a question. In `CoWork.h`, line 186, etc,

```
void Do(Function&& f, Args&&... args) { co.Do([=]() { ret = f(args...); }); }
```

shouldn't it be something like

```
void Do(Function&& f, Args&&... args) { co.Do([=]() { ret = f(std::forward<Args>(args)...); }); }
```

for perfect forwarding to actually work?

Follow up [https://www.ultimatepp.org/forums/index.php?t=msg&goto=6\\_0833&#msg\\_60833](https://www.ultimatepp.org/forums/index.php?t=msg&goto=6_0833&#msg_60833)

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Wed, 18 Sep 2024 15:54:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Yes, I have the same problem. That's weird.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Wed, 18 Sep 2024 16:08:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

mirek wrote on Wed, 18 September 2024 05:49Lance wrote on Tue, 17 September 2024 17:34  
The current upp codebase works fine with std=c++20, except the warning: implicit capture of 'this' with a capture default of '=' is deprecated [-Wdeprecated-this-capture] issue. This can be fixed without much effort and without affecting the aimed standard of c++17. I was wondering if you would consider accepting it if I make a pull request for that.

Nope, cannot do that with C++17. Requires [=, this], which is C++20 feature. And, frankly, while [=] capturing 'this' is perhaps confusing, everybody is used to it by now so to require longer code for the same thing is at this point is kind of pointless... But when we go C++20, we shall do it anyway. In 3 years...

Thank you for the feedback. At least it doesn't affect my using c++20. For now, I will just suppress the warnings.

We can predict moving to c++20 is going to be really smooth in the future. c++23 however will break many String interfaces. That's something we won't need to worry in like 6 years

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Wed, 18 Sep 2024 16:09:51 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Lance wrote on Wed, 18 September 2024 18:08c++23 however will break many String interfaces.

Oops, can you post links why?

Mirek

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Wed, 18 Sep 2024 16:54:28 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

with clang  
when compiler flags set to -std=c++23 -Wno-logical-op-parentheses

we have multiple errors with messages like

Quote:

```
/home/lance/upp.src/uppsrc/Core/String.h (556): error: non-const lvalue reference to type 'String'  
cannot bind to a temporary of type 'String'
```

And the relevant source is

```
force_inline String& operator<<(String&& s, const char *x)  
{  
    s.Cat(x, strlen__(x));  
    return s;  
}
```

Now c++23 is rejecting code like these.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Wed, 18 Sep 2024 19:56:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Lance wrote on Wed, 18 September 2024 18:54with clang  
when compiler flags set to -std=c++23 -Wno-logical-op-parentheses

we have multiple errors with messages like

Quote:

```
/home/lance/upp.src/uppsrc/Core/String.h (556): error: non-const lvalue reference to type 'String'  
cannot bind to a temporary of type 'String'
```

And the relevant source is

```
force_inline String& operator<<(String&& s, const char *x)  
{  
    s.Cat(x, strlen__(x));  
    return s;  
}
```

Now c++23 is rejecting code like these.

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Thu, 19 Sep 2024 02:47:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Fantastic. String are fixed. Now a simple Core console compiles with only one error.

Quote:

```
/home/lance/upp.src/uppsrc/Core/Stream.cpp (1226): error: no viable conversion from returned value of type 'StringBuffer' to function return type 'String'
```

And it points to Line 1226 of Core/Stream.cpp

```
String LoadFile(const char *filename) {
    FindFile ff(filename);
    if(ff && ff.IsFile()) {
#ifdef PLATFORM_POSIX
        if(ff.GetLength() == 0) { // handle special cases like /proc/...
            int fd = open(filename,O_RDONLY);
            if(fd >= 0) {
                const int CHUNK = 32768;
                StringBuffer s;
                for(;;) {
                    int n = s.GetCount();
                    s.SetCount(n + CHUNK);
                    int len = read(fd, ~s + n, CHUNK);
                    if(len != CHUNK) {
                        if(len >= 0)
                            s.SetCount(n + len);
                        close(fd);
                        return s; <---This Line
```

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Thu, 19 Sep 2024 07:53:54 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Sorry, only fixed Win32... Now MacOS and Linux should be fixed as well.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [coolman](#) on Thu, 19 Sep 2024 09:09:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Mirek,

Would it be possible to clean up these annoying warnings?

```
uppsrc/RichText/ParaPaint.cpp (468): warning: unused variable 'pe' [-Wunused-variable]
uppsrc/ide/IconDes/Event.cpp (102): warning: unused variable 'c' [-Wunused-variable]
uppsrc/ide/IconDes/IconDes.cpp (130): warning: unused variable 'isz' [-Wunused-variable]
uppsrc/ide/IconDes/ImlFile.cpp (272): warning: unused function 'PutOctalString'
[-Wunused-function]
uppsrc/ide/SelectPkg.cpp:661:3: warning: misleading indentation; statement is not part of the
previous 'if' [-Wmisleading-indentation]
 661 |         for(int i = 0; i < dlg.list.GetCount(); i++)
      |         ^
uppsrc/ide/SelectPkg.cpp:658:2: note: previous statement is here
 658 |     if(n)
      |     ^
1 warning generated.
```

Thank you, Radek

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Thu, 19 Sep 2024 09:54:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

coolman wrote on Thu, 19 September 2024 11:09: Hi Mirek,

Would it be possible to clean up these annoying warnings?

```
uppsrc/RichText/ParaPaint.cpp (468): warning: unused variable 'pe' [-Wunused-variable]
uppsrc/ide/IconDes/Event.cpp (102): warning: unused variable 'c' [-Wunused-variable]
uppsrc/ide/IconDes/IconDes.cpp (130): warning: unused variable 'isz' [-Wunused-variable]
uppsrc/ide/IconDes/ImlFile.cpp (272): warning: unused function 'PutOctalString'
[-Wunused-function]
uppsrc/ide/SelectPkg.cpp:661:3: warning: misleading indentation; statement is not part of the
previous 'if' [-Wmisleading-indentation]
 661 |         for(int i = 0; i < dlg.list.GetCount(); i++)
      |         ^
uppsrc/ide/SelectPkg.cpp:658:2: note: previous statement is here
 658 |     if(n)
      |     ^
```

1 warning generated.

Thank you, Radek

Is this GCC?

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Thu, 19 Sep 2024 12:27:13 GMT  
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Thu, 19 September 2024 03:53 Sorry, only fixed Win32... Now MacOS and Linux should be fixed as well.

The above problem is fixed. Now we are almost sure that U++ will not hindering people who wants to use the most recent language features.

When try to build UPP/AllForI18N, with CLANG+c++23, I got the following errors:

Quote:/home/lance/upp.src/uppsrc/MySql/MySql.h (9): fatal error: 'mysql.h' file not found

And

Quote:

/home/lance/upp.src/uppsrc/plugin/dbf/dbf.cpp (198): error: no viable conversion from returned value of type 'StringBuffer' to function return type 'String'

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [coolman](#) on Thu, 19 Sep 2024 13:05:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Thu, 19 September 2024 11:54 coolman wrote on Thu, 19 September 2024 11:09 Hi Mirek,

Would it be possible to clean up these annoying warnings?

uppsrc/RichText/ParaPaint.cpp (468): warning: unused variable 'pe' [-Wunused-variable]  
uppsrc/ide/IconDes/Event.cpp (102): warning: unused variable 'c' [-Wunused-variable]  
uppsrc/ide/IconDes/IconDes.cpp (130): warning: unused variable 'isz' [-Wunused-variable]  
uppsrc/ide/IconDes/ImlFile.cpp (272): warning: unused function 'PutOctalString' [-Wunused-function]

uppsrc/ide/SelectPkg.cpp:661:3: warning: misleading indentation; statement is not part of the previous 'if' [-Wmisleading-indentation]

```
661 |         for(int i = 0; i < dlg.list.GetCount(); i++)
    |         ^
uppsrc/ide/SelectPkg.cpp:658:2: note: previous statement is here
```

```
658 |     if(n)
    |     ^
1 warning generated.
```

Thank you, Radek

Is this GCC?

No, it is clang --version

Ubuntu clang version 17.0.6 (++)20231208085846+6009708b4367-1~exp1~20231208085949.74)

Target: x86\_64-pc-linux-gnu

Thread model: posix

InstalledDir: /usr/bin

---

Subject: Re: New release 2024 "alpha phase"

Posted by [mirek](#) on Fri, 20 Sep 2024 07:42:33 GMT

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

---

coolman wrote on Thu, 19 September 2024 15:05mirek wrote on Thu, 19 September 2024 11:54coolman wrote on Thu, 19 September 2024 11:09Hi Mirek,

Would it be possible to clean up these annoying warnings?

uppsrc/RichText/ParaPaint.cpp (468): warning: unused variable 'pe' [-Wunused-variable]

uppsrc/ide/IconDes/Event.cpp (102): warning: unused variable 'c' [-Wunused-variable]

uppsrc/ide/IconDes/IconDes.cpp (130): warning: unused variable 'isz' [-Wunused-variable]

uppsrc/ide/IconDes/ImlFile.cpp (272): warning: unused function 'PutOctalString' [-Wunused-function]

uppsrc/ide/SelectPkg.cpp:661:3: warning: misleading indentation; statement is not part of the previous 'if' [-Wmisleading-indentation]

```
661 |         for(int i = 0; i < dlg.list.GetCount(); i++)
    |         ^
uppsrc/ide/SelectPkg.cpp:658:2: note: previous statement is here
```

```
658 |     if(n)
    |     ^
1 warning generated.
```

Thank you, Radek

Is this GCC?

No, it is clang --version

Ubuntu clang version 17.0.6 (++)20231208085846+6009708b4367-1~exp1~20231208085949.74)

Target: x86\_64-pc-linux-gnu

Thread model: posix

InstalledDir: /usr/bin

Fixed. BTW, how is your build method CLANG settings? I have found that I have these warnings off by default... (looking for possibility to setup build method with these automatically)

---

Subject: Re: New release 2024 "alpha phase"

Posted by [coolman](#) on Fri, 20 Sep 2024 13:26:30 GMT

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

---

mirek wrote on Fri, 20 September 2024 09:42coolman wrote on Thu, 19 September 2024 15:05mirek wrote on Thu, 19 September 2024 11:54coolman wrote on Thu, 19 September 2024 11:09Hi Mirek,

Would it be possible to clean up these annoying warnings?

uppsrc/RichText/ParaPaint.cpp (468): warning: unused variable 'pe' [-Wunused-variable]

uppsrc/ide/IconDes/Event.cpp (102): warning: unused variable 'c' [-Wunused-variable]

uppsrc/ide/IconDes/IconDes.cpp (130): warning: unused variable 'isz' [-Wunused-variable]

uppsrc/ide/IconDes/ImlFile.cpp (272): warning: unused function 'PutOctalString'

[-Wunused-function]

uppsrc/ide/SelectPkg.cpp:661:3: warning: misleading indentation; statement is not part of the previous 'if' [-Wmisleading-indentation]

```
661 |         for(int i = 0; i < dlg.list.GetCount(); i++)
```

```
    |         ^
```

uppsrc/ide/SelectPkg.cpp:658:2: note: previous statement is here

```
658 |     if(n
```

```
    |     ^
```

1 warning generated.

Thank you, Radek

Is this GCC?

No, it is clang --version

Ubuntu clang version 17.0.6 (++)20231208085846+6009708b4367-1~exp1~20231208085949.74)

Target: x86\_64-pc-linux-gnu

Thread model: posix

InstalledDir: /usr/bin

Fixed. BTW, how is your build method CLANG settings? I have found that I have these warnings off by default... (looking for possibility to setup build method with these automatically)

I used release build to build theIDE.

## File Attachments

---

1) [build\\_method.png](#), downloaded 366 times

Build methods

- Method
- CLANG**
- MUSL
- GCC

Builder: CLANG

Compiler name clang++

External debugger lldb

Common options

Common C++ options -std=c++17 -Wno-logical-op-parentheses

Common C options

Common link options

Common fixed flags

Debug mode defaults

Default debug info level: Full

Use BLITZ

Debug options

-O0

Debug fixed flags

Debug link options

Release mode defaults

Use BLITZ

Release options

-O2 -ffunction-sections -fdata-sections -fstack-protect

Release fixed flags

Release link options

-Wl,--gc-sections,-z,now,-z,relro

Allow precompiled headers

Disable BLITZ

PATH - executable directories

INCLUDE directories

LIB directories

Lock link mode

Script file:

Store all target files in the same directory

Set as default

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Sat, 21 Sep 2024 12:41:39 GMT

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

---

Lance wrote on Thu, 19 September 2024 14:27  
03:53 Sorry, only fixed Win32... Now MacOS and Linux should be fixed as well.

The above problem is fixed. Now we are almost sure that U++ will not be hindering people who want to use the most recent language features.

When I try to build UPP/AllFor18N, with CLANG+c++23, I got the following errors:

Quote: /home/lance/upp.src/uppsrc/MySQL/MySQL.h (9): fatal error: 'mysql.h' file not found

And

Quote:

/home/lance/upp.src/uppsrc/plugin/dbf/dbf.cpp (198): error: no viable conversion from returned value of type 'StringBuffer' to function return type 'String'

Should be fixed. I have also added C++23 compilation to nightly testing routine and fixed all remaining problems - all of uppsrc should now work....

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Sat, 21 Sep 2024 21:31:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thank you, Mirek!

I am not sure if it's just me but, even with std=c++17 (or otherwise), the following problem persists  
Quote: /home/lance/upp/uppsrc/MySQL/MySQL.h (9): fatal error: 'mysql.h' file not found

It's not bothering me at all though.

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [mirek](#) on Sun, 22 Sep 2024 07:39:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Lance wrote on Sat, 21 September 2024 23:31  
Thank you, Mirek!

I am not sure if it's just me but, even with std=c++17 (or otherwise), the following problem persists  
Quote: /home/lance/upp/uppsrc/MySQL/MySQL.h (9): fatal error: 'mysql.h' file not found

It's not bothering me at all though.

Do you have mysql dev installed?

---

---

Subject: Re: New release 2024 "alpha phase"  
Posted by [Lance](#) on Sun, 22 Sep 2024 11:11:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Oh I see. That's why. Thanks.

---