Subject: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Fri, 30 Oct 2020 17:20:32 GMT

View Forum Message <> Reply to Message

if such http.buffer in debug (with multiple FreeFree_etc) means memory leaks?.. what does it mean?

I'm using mingw32 compiler v.9.3 & getting memory leaks in the situation described in the topic name, but when using mingw_ delivered as a part of U++ v.13664 -- memory not leaking (though another my stuff gives some mistakes)... what explanations can be (for mingw 9.3)?.. and what to do?.. or should I do everything in main thread - concerning loading several urls in asynchronous mode - is it the way to avoid memory leaks?... (:? just don't want to refactor for single-thread if the problem can really be in something else)

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by Klugier on Sat, 31 Oct 2020 10:18:16 GMT

View Forum Message <> Reply to Message

Hello,

Could you post test case for this (optimally simplify version)? Without the code it will be difficult to diagnose what is causing the problem.

Klugier

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Sat, 31 Oct 2020 10:38:35 GMT

View Forum Message <> Reply to Message

Klugier wrote on Sat, 31 October 2020 11:18Without the code it will be difficult to diagnose what is causing the problem.
well... it can take time... I just saw such variable in debugger (with multiple FreeFreeFree...) when I had a memory leak... the first question is just about - does it really mean that the memory is being allocated, but not freed?..
p.s.
to extract a piece of code will be really not so easy - will took time... I will see...
just I saw some preventions in forum, that openSSL sometimes gives memory leaks when working in MT-environment (in my case in secondary thread, though when I do loop in main-gui-thread - no problems - already was tested)...
 - if this problem can occur in v.13664-32x-windows or it depends on compiler (mingw or MSVC)?.. - I'm not sure, that the whole code is needed - especially if I'm restricted in using just last 32x-version of U++ in windows with mingw-compiler (optimized to 9.3 version in Build_Method_Settings & I've got away everything else that was there - perhaps I've lost something important?)

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by Klugier on Sat, 31 Oct 2020 14:02:29 GMT

Hello,

Did you try to debug memory leaks? It will show you exactly in which place the leak is. The all information about detecting memory leaks and marking false positives you can find here.

Please noticed that today I updated this article significantly, so not all information will be there. Visit this site tomorrow and all new information should be there. Alternatively you could update to the latest SVN version and read new documentation page inside TheIDE or just read the raw data in commit content available on GitHub (Click on Load Diff for obtaining content).

Klugier

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Sat, 31 Oct 2020 15:01:24 GMT

Klugier wrote on Sat, 31 October 2020 15:02
Did you try to debug memory leaks?
I do have such main
GUI_APP_MAIN
{
#ifdef _DEBUG
 StdLogSetup(LOG_COUT|LOG_FILE);
 HttpRequest::Trace();
#endif

   MyAppWindow().Run();
}
and getting such LOG (under spoiler)
Toggle SpoilerQuote:Heap leaks detected:

--memory-breakpoint__ 84381 : Memory at 0x041bccb0, size 0x10C = 268
   +0 0x041BCCB0 0C 01 00 00 00 00 00 00 EB 75 72 F2 5B 97 AD 42   ....ur.[..B
   +16 0x041BCCC0 B8 40 5C F3 9C D6 60 6F 04 E5 D8 FA 22 F2 31 28   .@\...`o....".1(
   +32 0x041BCCD0 B5 87 19 22 03 B2 77 89 4C 0E 45 06 17 99 E8 44   ..."..w.L.E....D
   +48 0x041BCCE0 AF D9 B4 B7 33 0F D4 D8 C7 93 90 9B E5 61 A1 B3   ....3........a..

--memory-breakpoint__ 84341 : Memory at 0x041bca70, size 0x10C = 268
   +0 0x041BCA70 0C 01 00 00 00 00 00 00 0B FA 4A B4 87 F2 DB 69   ....J....i
   +16 0x041BCA80 71 09 75 20 F5 41 BD 25 CE 1E 20 60 45 A0 CE 11   q.u .A.%.. `E...
   +32 0x041BCA90 5E F8 BD 15 4A 18 8E 1D AF 2C E7 AC 28 DE 3C C5   ^...J....,..(.<.
   +48 0x041BCAA0 59 D7 49 E5 AC 96 F4 C0 5F 8E 9F DA 1A 2E 51 CB   Y.I....._.....Q.

```
--memory-breakpoint__ 15763 : Memory at 0x041be470, size 0x4C = 76
    +0 0x041BE470 4C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    L...............
   +16 0x041BE480 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +32 0x041BE490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +48 0x041BE4A0 00 00 00 00 00 00 00 00 46 72 65 65 46 72 65 65    ........FreeFree

--memory-breakpoint__ 15762 : Memory at 0x041c0a10, size 0x2C = 44
    +0 0x041C0A10 2C 00 00 00 00 00 00 00 78 E4 1B 04 00 00 00 00    ,.......x.......
   +16 0x041C0A20 00 00 00 00 00 00 00 00 00 00 00 00 00 30 00 00    .............0..
   +32 0x041C0A30 30 00 00 00 00 00 00 00 00 00 00 00             0..........

--memory-breakpoint__ 15753 : Memory at 0x041bcb90, size 0x10C = 268
    +0 0x041BCB90 0C 01 00 00 00 00 00 00 03 02 01 00 07 06 05 04    ................
   +16 0x041BCBA0 0B 0A 09 08 0F 0E 0D 0C 13 12 11 10 17 16 15 14    ................
   +32 0x041BCBB0 1B 1A 19 18 1F 1E 1D 1C 9F C2 73 A5 98 C4 76 A1    ..........s...v.
   +48 0x041BCBC0 93 CE 7F A9 9C C0 72 A5 CD A8 51 16 DA BE 44 02    ..•...r...Q...D.

--memory-breakpoint__ 15752 : Memory at 0x041a64d0, size 0xAC = 172
    +0 0x041A64D0 AC 00 00 00 00 00 00 00 C0 3A A8 00 00 00 00 00    .........:......
   +16 0x041A64E0 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +32 0x041A64F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +48 0x041A6500 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................

--memory-breakpoint__ 15751 : Memory at 0x041a6410, size 0xAC = 172
    +0 0x041A6410 AC 00 00 00 00 00 00 00 C0 3A A8 00 00 00 00 00    .........:......
   +16 0x041A6420 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +32 0x041A6430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +48 0x041A6440 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................

--memory-breakpoint__ 15750 : Memory at 0x041c24d0, size 0x12C = 300
    +0 0x041C24D0 2C 01 00 00 00 00 00 00 00 00 00 00 58 22 1C 04    ,...........X"..
   +16 0x041C24E0 00 00 00 00 8A 03 00 00 00 00 00 00 00 00 00 00    ................
   +32 0x041C24F0 00 00 00 00 18 0A 1C 04 00 01 00 00 00 00 01 00    ................
   +48 0x041C2500 20 00 00 00 FF FF FF 7F 10 00 00 00 FF FF FF 7F     ......•.......•

--memory-breakpoint__ 15746 : Memory at 0x041be4d0, size 0x4C = 76
    +0 0x041BE4D0 4C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    L...............
   +16 0x041BE4E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +32 0x041BE4F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
   +48 0x041BE500 00 00 00 00 00 00 00 00 46 72 65 65 46 72 65 65    ........FreeFree

--memory-breakpoint__ 15745 : Memory at 0x041c09d0, size 0x2C = 44
    +0 0x041C09D0 2C 00 00 00 00 00 00 00 D8 E4 1B 04 00 00 00 00    ,...............
   +16 0x041C09E0 00 00 00 00 00 00 00 00 00 00 00 00 00 30 00 00    .............0..
   +32 0x041C09F0 30 00 00 00 00 00 00 00 00 00 00 00             0..........

--memory-breakpoint__ 15736 : Memory at 0x041bc950, size 0x10C = 268
    +0 0x041BC950 0C 01 00 00 00 00 00 00 03 02 01 00 07 06 05 04    ................
```

```
  +16 0x041BC960 0B 0A 09 08 0F 0E 0D 0C 13 12 11 10 17 16 15 14      ................
  +32 0x041BC970 1B 1A 19 18 1F 1E 1D 1C 9F C2 73 A5 98 C4 76 A1      ..........s...v.
  +48 0x041BC980 93 CE 7F A9 9C C0 72 A5 CD A8 51 16 DA BE 44 02      ..•...r...Q...D.

--memory-breakpoint__ 15735 : Memory at 0x041a6350, size 0xAC = 172
   +0 0x041A6350 AC 00 00 00 00 00 00 00 C0 3A A8 00 00 00 00 00      .........:......
  +16 0x041A6360 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  +32 0x041A6370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  +48 0x041A6380 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................

--memory-breakpoint__ 15734 : Memory at 0x041a6290, size 0xAC = 172
   +0 0x041A6290 AC 00 00 00 00 00 00 00 C0 3A A8 00 00 00 00 00      .........:......
  +16 0x041A62A0 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  +32 0x041A62B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  +48 0x041A62C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................

--memory-breakpoint__ 15733 : Memory at 0x041c2390, size 0x12C = 300
   +0 0x041C2390 2C 01 00 00 00 00 00 00 00 00 00 00 58 22 1C 04      ,...........X"..
  +16 0x041C23A0 00 00 00 00 8A 03 00 00 00 00 00 00 00 00 00 00      ................
  +32 0x041C23B0 00 00 00 00 D8 09 1C 04 00 01 00 00 00 00 01 00      ................
  +48 0x041C23C0 20 00 00 00 FF FF FF 7F 10 00 00 00 FF FF FF 7F      ......•.......•

--memory-breakpoint__ 15683 : Memory at 0x041c04d0, size 0x2C = 44
   +0 0x041C04D0 2C 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00      ,...............
  +16 0x041C04E0 01 00 00 00 46 72 65 65 46 72 65 65 46 72 65 65      ....FreeFreeFree
  +32 0x041C04F0 46 72 65 65 46 72 65 65 46 72 65 65                  FreeFreeFree

--memory-breakpoint__ 15682 : Memory at 0x041b9210, size 0x1AC = 428
   +0 0x041B9210 AC 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  +16 0x041B9220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  +32 0x041B9230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
  +48 0x041B9240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
****************** PANIC: Memory leaks detected! (final check)
```

but didn't know what to do with it :blush: - taking into consideration this part:
Quote:Of course, the only problem with this is that the order of allocations must be exactly the same in the "test run". Which is often not, but usually it is possible to arrange it so after spending a couple of nights desperately looking for the source of leak in the code using other methods...
... while using ONLY U++ & neither VisualStudio tools nor Deleaker for Qt - it was problematic for me to deal with this LOG's info... therefore I have only debugger in my disposal & its FreeFreeFree_etc in one variable & my supposition that it concerns the leak?.. by the way original code, though not leaking is in examples in u++ (as in my screen was) & when using it not in main, but in secondary thread - I'm getting memory leaks (as I already mentioned)... therefore I supposed that httpRequest can give memory leak, but in ST-app these leaks are being hidden somehow, though in MT-app are being revealed by assertion... IMHO... if async loading takes place in secondary thread... BTW it's not a Server, just some requests to some html_pages (multiple)...
Klugier wrote on Sat, 31 October 2020 15:02

visit this site tomorrow and all new information should be there.

:) ok - will wait till tomorrow - thank you!
Klugier wrote on Sat, 31 October 2020 15:02
Alternatively you could update to the latest SVN version and read new documentation page inside TheIDE

I do not see any updater in the ide itself... is it possible only from the git? - I am not connected to it anyway...
Klugier wrote on Sat, 31 October 2020 15:02
or just read the raw data in commit content available on GitHub (Click on Load Diff for obtaining content).

have got... thank you!
but my results you can see under spoiler in this message above here - until I'll find what to do with it from your paper... :roll: because I don't know yet... and you are not telling about openSSL behavior in my situation described

---

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Sat, 31 Oct 2020 16:34:40 GMT
View Forum Message <> Reply to Message

Oh, I really have already got it into my U++ - thanks a lot! - for detailed description, code examples & even pictures == that is really very informative! - will learn... thank you for spending time to do such update into Help section
p.s.
now I even see that updating to the latest SVN is very simple :) (just copy-paste needed files)

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by mirek on Mon, 02 Nov 2020 10:57:08 GMT
View Forum Message <> Reply to Message

JeyCi wrote on Sat, 31 October 2020 11:38Klugier wrote on Sat, 31 October 2020 11:18Without the code it will be difficult to diagnose what is causing the problem.
well... it can take time... I just saw such variable in debugger (with multiple FreeFreeFree...) when I had a memory leak... the first question is just about - does it really mean that the memory is being allocated, but not freed?..

If you see pointer pointing to "FreeFreeFree....", it most likely means that it points to the memory that was already deallocated.

In debug mode, U++ allocator fills the memory with this pattern on deallocation. Then when the same block is allocated again, it checks that the pattern is unchanged. If it is not, it would means

something has written to that memory after it was deallocated, which is a sign of dangling pointer problem. In that case the app is aborted (in debug).

Mirek

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Tue, 03 Nov 2020 07:07:33 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 31 October 2020 15:02It will show you exactly in which place the leak is.
thank you all for your replies! - I've got the place where it leaks:
it leaks here when httpRequests are being done in secondary thread (not gui-thread) - when using --memory-breakpoint__ [num detected before] - myApp stops here in debug

? can it be connected with OpenSSL, perhaps, as external library is considered to be?
or just the problem of my app_design?..
perhaps, declaring Vector <urls> as thread-local prior to start of thread can help?..
but frankly speaking I intend to consider the error arising just because of switching-context that takes place in MT-environment & not freeing http.buffer... though I added w.http.ClearContent(); in the code...
or perhaps really http.buffer for each request should have thread-local storage-duration?...
p.s.
(btw I'm gathering the Vector of links in the same secondary thread - not transferring it from main thread)
Anyway, thank you, - now I see exact place where it happens

```
File Attachments
1) 03.11.jpg, downloaded 584 times
```

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by mirek on Tue, 03 Nov 2020 08:08:59 GMT
View Forum Message <> Reply to Message

JeyCi wrote on Tue, 03 November 2020 08:07Klugier wrote on Sat, 31 October 2020 15:02It will show you exactly in which place the leak is.
thank you all for your replies! - I've got the place where it leaks:
it leaks here when httpRequests are being done in secondary thread (not gui-thread) - when using --memory-breakpoint__ [num detected before] - myApp stops here in debug

? can it be connected with OpenSSL, perhaps, as external library is considered to be?
or just the problem of my app_design?..
perhaps, declaring Vector <urls> as thread-local prior to start of thread can help?..
but frankly speaking I intend to consider the error arising just because of switching-context that takes place in MT-environment & not freeing http.buffer... though I added w.http.ClearContent(); in the code...

or perhaps really http.buffer for each request should have thread-local storage-duration?...
p.s.
(btw I'm gathering the Vector of links in the same secondary thread - not transferring it from main thread)
Anyway, thank you, - now I see exact place where it happens

The interesting info you should have sent is the backtrace. The best way is (after it stops on memory breakpoint) Debug / Copy backtrace of all threads.

HttpRequest should not leak (of course, it is always possible there is a bug). Any chance you are doing something like early exit from the thread?

Ah, and maybe the most important: Are you using Upp::Thread? Or are how do you start the thread?

Mirek

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Tue, 03 Nov 2020 12:40:53 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 03 November 2020 09:08 The best way is (after it stops on memory breakpoint) Debug / Copy backtrace of all threads.
? what do you mean "backtrace" - is it this dropdown
[
mirek wrote on Tue, 03 November 2020 09:08Any chance you are doing something like early exit from the thread?
seems not - I'm simplifying testing case :)
mirek wrote on Tue, 03 November 2020 09:08
Ah, and maybe the most important: Are you using Upp::Thread? Or are how do you start the thread?

member Thread work; - in class MyAppWindow : public WithMyAppWindowLayout<TopWindow>
starting work from here:
void MyAppWindow:: btnStart_Click()
{
    work.Run ([=]{ WorkerThread(); });
}
I thought that native for U++ Thread is OK & I do not need creating my own wrapper_class for executing my thread - I like opportunities U++ gives with its Thread class... or do you think I'd better create my separate wrapper_class to work with Thread?.. something like RAII-style...
though I don't need any special ways of synchronization - am just using THISFN for interaction with Gui, using GuiLock/Call in these functions...
===
So, here:
    MemoryIgnoreLeaksBegin() ;
    w.http.Do();

MemoryIgnoreLeaksEnd();
- with thus it is OK when being executed Workerthread() function...
or do you think it is better to create a separate class to wrap the Thread?..
===
BTW the situation still differs depending on chosen compiler - as I have written - with
MINGW_built_in_U++ - OK working... but with my newer mingw_9_3 in Build Method I do have
leak in the place I've shown in the code (switching-on MemoryIgnore in that line - function works
good)...
because I deleted everything in BuildMethod from U++ but made settings (path, include, lib) only
for foreign mingw (loaded with msys2) - perhaps I have lost something important in my settings
(when removed native stuff)?
though in msys2 I have even done separate installation additionally to mingw itself:
Installation:pacman -S openssl
===
oh, mirek - I think I can suppose the reason - msys2 is also no more supporting x32 from May
2020... you was right - most modern CPUs support the possibility to increase RAM-size... really...
Anyway it seems I now do know how to find memory_leak in U++ - no other foreign tools needed
-- Thank you...
and the testing code still works without memory-leaks in U++-v.13664 - the version in which
mingw already build_in

File Attachments
1) 03.11_2.jpg, downloaded 569 times

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by mirek on Tue, 03 Nov 2020 12:57:03 GMT
View Forum Message <> Reply to Message

You should have invoked that menu command I have suggested, that would put the whole
backtrace snapshot on the clipboard that you could post here (e.g. as attachement). Much better
than screenshots... :)

Using Thread is fine! I was actually afraid that you are not using it, that would lead to leaks. You
need to use Upp::Thread as it contains some allocator related stuff to release some allocation
caches...

Anyway, from the very sparse info I am getting, I am starting to think that you migh not be locking
things properly when communicating between threads. I would guess that 'showDone' is
something that you call from the thread to show progress in GUI. Do you lock things propertly?
E.g. with GuiLock __?

Mirek

Subject: Re: httpRequests in secondary non-gui-main Thread

Posted by mirek on Tue, 03 Nov 2020 12:59:42 GMT
View Forum Message <> Reply to Message

JeyCi wrote on Tue, 03 November 2020 13:40mirek wrote on Tue, 03 November 2020 09:08 The best way is (after it stops on memory breakpoint) Debug / Copy backtrace of all threads.
? what do you mean "backtrace" - is it this dropdown

Once it stops, click "Debug" menu and choose "Copy backtrace of all threads". It will put the backtrace onto clipboard which you can then paste here.

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Tue, 03 Nov 2020 14:07:14 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 03 November 2020 13:57I would guess that 'showDone' is something that you call from the thread to show progress in GUI. Do you lock things propertly? E.g. with GuiLock __?

no it isn't... I just didn't care about the name of the method... but as you can notice this method is being got from another class (asyncHttp)... so I'm starting the WorkerThread as I have shown above... & I separated the loading itself to the separate class that takes vector done in WorkerThread() as Constructor's parameter & several THISFNs as parameters of the method showDone (but its real purpose is exactly loading stuff & returning results to functions of gui_class, in which I of course use GuiLock & Call([=]{this->finished.SetArray(i, va)})...
well but it was prior design, then I of course refactored for not returning results to gui... just from the beginning I wanted to see what is returning - but in the refactored code everything goes to SaveFile(n,res)... I had finally decreased the quantity of synchronization points to zero now... just have now taken for the test the shorter code - to reduce it even more for testing purposes (but it just appered to be the code still with synchronization)...
but the problem in all my step_by_step dev.process is the same - the line I've taken in MemoryIgnore... perhaps you're right - perhaps I shouldn't do loading stuff in the separate class?.. not knowing how to make this class thread-safe?.. perhaps it is not thread-save in my situation described?
mirek wrote on Tue, 03 November 2020 13:57You should have invoked that menu command I have suggested, that would put the whole backtrace snapshot on the clipboard that you could post here (e.g. as attachement).
 :o ah, I see now... ok when I will shorten the whole code even more - perhaps... in order not to provide unneeded info... (if given description doesn't clarify the surrounding of the problem)

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by mirek on Tue, 03 Nov 2020 14:46:06 GMT
View Forum Message <> Reply to Message

Well, really, I would like to help, but I am completely in the dard about the code... I will stop

guessing now and wait for simplified testcase.

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Tue, 03 Nov 2020 17:46:12 GMT
View Forum Message <> Reply to Message

You already gave me all key points to check in my code - thank you - if it is working with mingw_that_is build in U++ - then will be ok anyway... just it started to work when I started to shorten it... I will check everything again taking into consideration all your advices - :) it can appear to be quicker then to shorten the code & leaving the error...
p.s.
for any case - I've read this topic - Win32 TLS & this - release 2019.2 (Aug) - workaround for Mingw TLS performance issue - to remember whether I have this in v.13664 - if I will need it... or to change my design... just for info...
but I will check thoroughly how do I pass parameters to the Class doing loading stuff & how do I make destruction of its objects in order then to exit thread-function correctly

---

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Fri, 06 Nov 2020 02:57:27 GMT
View Forum Message <> Reply to Message

JeyCi wrote on Tue, 03 November 2020 18:46release 2019.2[/url] (Aug) - workaround for Mingw TLS performance issue - to remember whether I have this in v.13664
exactly the version I use - 13664

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Fri, 06 Nov 2020 12:03:48 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 03 November 2020 13:57You put the whole backtrace snapshot on the clipboard that you could post here (e.g. as attachement). Much better than screenshots...
seeing under spoiler
Toggle Spoiler

Thread: 1

Upp::Panic (msg=0xa5ddcc <Upp::RECT64_V+100> "Memory leaks detected! (final check)") at D:/TODO/upp 13664/upp/uppsrc/Core/Util.cpp:119
Upp::Heap::AssertLeaks (b=false) at D:\TODO\upp 13664\upp\uppsrc\Core\heap.cpp:230
Upp::Heap::AuxFinalCheck () at D:\TODO\upp 13664\upp\uppsrc\Core\heap.cpp:245
Upp::MemoryDumpLeaks () at D:\TODO\upp 13664\upp\uppsrc\Core\heapdbg.cpp:243
MemDiagCls::~MemDiagCls (this=0xdb094c <sMemDiagHelper>, __in_chrg=<optimized out>) at D:\TODO\upp 13664\upp\uppsrc\Core\heapdbg.cpp:280
13664\upp\uppsrc\Core\heapdbg.cpp:283

---

msvcrt!_initterm_e () from C:\Windows\System32\msvcrt.dll
msvcrt!exit () from C:\Windows\System32\msvcrt.dll
        D:/mingwbuild/mingw-w64-crt-git/src/mingw-w64/mingw-w64-crt/ crt/crtexe.c:340
KERNEL32!BaseThreadInitThunk () from C:\Windows\System32\kernel32.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll

----
Thread: 2

ntdll!KiFastSystemCallRet () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!ZwWaitForWorkViaWorkerFactory () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDeleteBoundaryDescriptor () from C:\Windows\SYSTEM32\ntdll.dll
KERNEL32!BaseThreadInitThunk () from C:\Windows\System32\kernel32.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll

----
Thread: 3

ntdll!KiFastSystemCallRet () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!ZwWaitForWorkViaWorkerFactory () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDeleteBoundaryDescriptor () from C:\Windows\SYSTEM32\ntdll.dll
KERNEL32!BaseThreadInitThunk () from C:\Windows\System32\kernel32.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll

----
Thread: 4

ntdll!KiFastSystemCallRet () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!ZwWaitForWorkViaWorkerFactory () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDeleteBoundaryDescriptor () from C:\Windows\SYSTEM32\ntdll.dll
KERNEL32!BaseThreadInitThunk () from C:\Windows\System32\kernel32.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll

----
Thread: 5

ntdll!KiFastSystemCallRet () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!ZwWaitForMultipleObjects () from C:\Windows\SYSTEM32\ntdll.dll
WaitForMultipleObjectsEx () from C:\Windows\System32\KernelBase.dll
combase!CoCreateGuid () from C:\Windows\System32\combase.dll
combase!CoGetContextToken () from C:\Windows\System32\combase.dll
combase!CoCreateGuid () from C:\Windows\System32\combase.dll
combase!CoGetStdMarshalEx () from C:\Windows\System32\combase.dll
KERNEL32!BaseThreadInitThunk () from C:\Windows\System32\kernel32.dll

ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll

----
Thread: 6

ntdll!KiFastSystemCallRet () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!ZwWaitForWorkViaWorkerFactory () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDeleteBoundaryDescriptor () from C:\Windows\SYSTEM32\ntdll.dll
KERNEL32!BaseThreadInitThunk () from C:\Windows\System32\kernel32.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll

----
Thread: 30

ntdll!KiFastSystemCallRet () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!ZwRemoveIoCompletion () from C:\Windows\SYSTEM32\ntdll.dll
Tcpip6_WSHStringToAddress () from C:\Windows\system32\mswsock.dll
KERNEL32!BaseThreadInitThunk () from C:\Windows\System32\kernel32.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll
ntdll!RtlDestroyQueryDebugBuffer () from C:\Windows\SYSTEM32\ntdll.dll

attempt to compile under mingw_9_3 (from msys2) seems to leak somewhere in System32??..
perhaps really the problem is in external lib... if I'm seeing correctly
p.s.
though it is even strange to see mingw-w64... as for my win32-system I probably need
i686-w64-mingw32 from msys2 folder...
p.p.s
:) though a little bit having refactored the whole code (not shortening) with new knowledge from
this topic - I do not have leak with mingw-build-in-U++... seems OK

File Attachments
1) 06.11.jpg, downloaded 423 times

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by JeyCi on Mon, 07 Dec 2020 07:01:59 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 31 October 2020 15:02 Alternatively you could update to the latest SVN
version and read new documentation page inside TheIDE
BTW, for info - a detailed, step by step, description how to build the IDE from the fresh SVN
sources (the part at the bottom of page)

Subject: Re: httpRequests in secondary non-gui-main Thread
Posted by mirek on Fri, 11 Dec 2020 14:48:42 GMT

BTW,

 https://github.com/mstorsjo/llvm-mingw/releases/tag/20201020

llvm-mingw-20201020-msvcrt-i686.zip

is 32-bit version of compiler toolchain. Should work fine with 32-bit libraries we supply in nightly. Perhaps this might solve your issue?

Mirek