
Subject: FtpClient heap leak

Posted by [forlano](#) on Sat, 03 Aug 2024 10:17:34 GMT

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

Hello,

my app use the class FtpClient. Only today I got in DEBUG mode a "heap leak detected" on exit (see below) although it worked OK.

In release mode nothing happen and no log file is saved. Perhaps this is an old issue because on DEBUG I never used this class so far.

From the log file it seems depends by FtpClient.Connect that I use here

```
if(!ftp.Connect( FTPHOST, FTPUSER, FTPPASS, true)) {  
    Exclamation("Unable to connect!" + ftp.GetError());  
    return;  
}
```

Is this a know issue?

Clang, Windows, Upp 17185

Thanks,
Luigi

* C:\upp\out\MyApps\CLANGx64.Debug.Debug_Full.Gui.Noblitz\Vega10.exe 03.08.2024
12:08:11, user: Vegachess

FtpClient::Connect
//FtpClient::Connect

Heap leaks detected:

```
--memory-breakpoint__ 14010 : Memory at 0x00000206c2b75ae0, size 0x3C = 60  
+0 0x00000206C2B75AE0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31  
1111111111111111  
+16 0x00000206C2B75AF0 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65  
1111111111.eFree  
+32 0x00000206C2B75B00 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65  
FreeFreeFreeFree  
+48 0x00000206C2B75B10 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree  
  
--memory-breakpoint__ 14009 : Memory at 0x00000206c2b75a80, size 0x3C = 60  
+0 0x00000206C2B75A80 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31  
1111111111111111
```

```

+16 0x00000206C2B75A90 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75AA0 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75AB0 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14008 : Memory at 0x00000206c2b75a20, size 0x3C = 60
+0 0x00000206C2B75A20 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75A30 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75A40 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75A50 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14007 : Memory at 0x00000206c2b759c0, size 0x3C = 60
+0 0x00000206C2B759C0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B759D0 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B759E0 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B759F0 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14006 : Memory at 0x00000206c2b75960, size 0x3C = 60
+0 0x00000206C2B75960 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75970 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75980 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75990 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14005 : Memory at 0x00000206c2b75900, size 0x3C = 60
+0 0x00000206C2B75900 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75910 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75920 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75930 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14004 : Memory at 0x00000206c2b758a0, size 0x3C = 60
+0 0x00000206C2B758A0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B758B0 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B758C0 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65

```

```

FreeFreeFreeFree
+48 0x00000206C2B758D0 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14003 : Memory at 0x00000206c2b75840, size 0x3C = 60
+0 0x00000206C2B75840 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75850 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75860 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75870 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14002 : Memory at 0x00000206c2b757e0, size 0x3C = 60
+0 0x00000206C2B757E0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B757F0 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75800 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75810 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14001 : Memory at 0x00000206c2b75780, size 0x3C = 60
+0 0x00000206C2B75780 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75790 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B757A0 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B757B0 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 14000 : Memory at 0x00000206c2b75720, size 0x3C = 60
+0 0x00000206C2B75720 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75730 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75740 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75750 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 13999 : Memory at 0x00000206c2b756c0, size 0x3C = 60
+0 0x00000206C2B756C0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B756D0 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B756E0 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B756F0 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

```

```

--memory-breakpoint__ 13998 : Memory at 0x00000206c2b75660, size 0x3C = 60
+0 0x00000206C2B75660 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75670 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75680 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75690 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 13997 : Memory at 0x00000206c2b75600, size 0x3C = 60
+0 0x00000206C2B75600 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75610 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75620 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75630 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 13996 : Memory at 0x00000206c2b755a0, size 0x3C = 60
+0 0x00000206C2B755A0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B755B0 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B755C0 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B755D0 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 13995 : Memory at 0x00000206c2b75540, size 0x3C = 60
+0 0x00000206C2B75540 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B75550 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75560 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75570 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

--memory-breakpoint__ 13994 : Memory at 0x00000206c2b754e0, size 0x3C = 60
+0 0x00000206C2B754E0 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
1111111111111111
+16 0x00000206C2B754F0 31 31 31 31 31 31 31 31 31 31 00 65 46 72 65 65
1111111111.eFree
+32 0x00000206C2B75500 46 72 65 65 46 72 65 65 46 72 65 65 46 72 65 65
FreeFreeFreeFree
+48 0x00000206C2B75510 46 72 65 65 46 72 65 65 46 72 65 65          FreeFreeFree

```

*** TOO MANY LEAKS (1201) TO LIST THEM ALL

Subject: Re: FtpClient heap leak
Posted by [Oblivion](#) on Sat, 03 Aug 2024 10:40:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Luigi,

Just curious: Have you every tried the FTP package from upp-components? Upp::FtpClient is while ok, FTP comes with examples and docs (and with a ton of modern (as modern as FTP can be) features you might find useful in modern environments.

Best regards,
Oblivion

Subject: Re: FtpClient heap leak
Posted by [forlano](#) on Sat, 03 Aug 2024 10:50:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oblivion wrote on Sat, 03 August 2024 12:40Hello Luigi,

Just curious: Have you every tried the FTP package from upp-components?

Hi Oblivion,

not yet tried. Is there a reccomended way to import it in my package?

Best regards,
Luigi

Subject: Re: FtpClient heap leak
Posted by [Oblivion](#) on Sat, 03 Aug 2024 11:05:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

The package is a straightforward Upp package, all you need to do is add the package to your project.

But before doing that I recommend reading the api docs and playing with the provided examples.

You don't need to download upp-components, I am going to create a UppHub package. Hopefully, it'll be available from UppHub in a couple of days.

However, in the meantime you can try the below attached package (it includes the FTP package and its reference examples.)

Best regards,

Oblivion

File Attachments

1) [FTP_Package.zip](#), downloaded 241 times

Subject: Re: FtpClient heap leak

Posted by [forlano](#) on Sat, 03 Aug 2024 21:03:38 GMT

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

Oblivion wrote on Sat, 03 August 2024 13:05

However, in the meantime you can try the below attached package (it includes the FTP package and its reference examples.)

Best regards,

Oblivion

Hi Oblivion,

I am trying your package. I looked at the example but I cannot match the interface/method of plugin/Ftp I was used to. Now I have many complain from compiler. In particular:

1. after connect, how to enter in a given folder? I had `ftp.Cd(folder)`

I guess now it is: `.SetDir(folder)`

2. How can I transfer a file named myfile.txt and content hold in the String ff? I used

`ftp.Save( "myfile.txt", ff)`

3. How to close the connection?

I had `Close()` and now? I guess it is `.Disconnect()`

Thanks,

Luigi

Subject: Re: FtpClient heap leak

Posted by [Oblivion](#) on Sun, 04 Aug 2024 09:42:28 GMT

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

Hello Luigi,

Quote:1. after connect, how to enter in a given folder? I had `ftp.Cd(folder)`

```
bool Ftp::SetDir(const String& path)
```

Quote:

2. How can I transfer a file named myfile.txt and content hold in the String ff? I used

Data transfer methods:

String	Get(const String& path, bool ascii = false);
bool	Get(const String& path, Stream& s, bool ascii = false);
bool	Put(Stream& s, const String& path, bool ascii = false);
bool	Put(const String& s, const String& path, bool ascii = false);
bool	Append(Stream &s, const String& path, bool ascii = false);
bool	Append(const String& s, const String& path, bool ascii = false);

Quote: 3. How to close the connection?

You can use,

```
Disconnect();
```

Note that Disconnect() method is also called in the destructor, if it is not explicitly called.

Basic data transfer example:

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

```
using namespace Upp;
```

```
CONSOLE_APP_MAIN
```

```
{
    StdLogSetup(LOG_COUT|LOG_FILE);
    // Ftp::Trace();
```

```
    Ftp ftpclient;
    if(ftpclient.Timeout(60000).Connect("ftp://demo:password@test.rebex.net:21")) {
        auto f = ftpclient.Get("readme.txt", true);
        if(!ftpclient.IsError()) {
            RLOG(f);
            return;
        }
    }
```

```
}  
RLOG(ftpclient.GetErrorDesc());  
}
```

Best regards,
Oblivion

Subject: Re: FtpClient heap leak
Posted by [forlano](#) on Sun, 04 Aug 2024 19:32:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Oblivion,

I added your FTP package and it works. On exit in debug mode the heap leak disappeared.
I used this method for upload in the current directory

```
bool Put(const String& s, const String& path, bool ascii = false);
```

where s is the string content and path the filename associated with s.
I have not tried a filename with a path. I wonder if this path is absolute or relative to the current directory. Moreover if the path contains a not existing folder on the server is it created?

Thanks,
Luigi

Subject: Re: FtpClient heap leak
Posted by [Oblivion](#) on Sun, 04 Aug 2024 21:20:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Luigi,

While the file transfer protocol allows setting the current/working directory using the "cd" command (Ftp::SetDir(const String& path)), you are still able to use absolute paths.
So, yes, unless the server is configured otherwise, you can do that.

As for the creation of a non-existing folder by specifying the path of a file: Again, it depends on the server and its configuration.

Best regards,

Oblivion

Subject: Re: FtpClient heap leak
Posted by [Oblivion](#) on Sat, 31 Aug 2024 19:59:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Luigi,

FYI, Ftp package is added to UppHub. You can now download and update it using UppHub package manager via TheIDE.

Best regards,

Oblivion
