
Subject: How to close the websocket connection
Posted by [Tess](#) on Tue, 07 Nov 2017 10:37:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

I tried this: [https://www.ultimatepp.org/reference\\$AsyncWebSocket\\$en-us.html](https://www.ultimatepp.org/reference$AsyncWebSocket$en-us.html) example. But the connections do not close, because `IsClosed()` method doesn't return true.

My compiler: gcc version 5.4.0

Subject: Re: How to close the websocket connection
Posted by [shutalker](#) on Fri, 10 Nov 2017 06:16:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

I've tried to run the example and got the same problem.

I noticed that there were still 2 threads (server and client) after server had responded to the client's single request .

Then I added LOG macros in Client.cpp right after websocket's close method invocations, but they weren't appear in logs. I also tried to start client from another application with debugger, but TheIDE crashed after close method had been invoked. So I suppose that something is blocking program execution at the moment of invocation of close method.

I use upp sources from git: 97e1f20
Compiler: GCC 5.4.0

Subject: Re: How to close the websocket connection
Posted by [mirek](#) on Wed, 15 Nov 2017 07:57:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hopefully fixed.

Thanks for reporting and sorry for the trouble.

Subject: Re: How to close the websocket connection
Posted by [shutalker](#) on Thu, 23 Nov 2017 20:35:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for the reply!

Unfortunately, this didn't fix the problem. Close method is still crashing my apps. I've tried to enable WebSocket tracing on client and server and I've noticed that server hadn't recieved close

message from the client although there was a message about sending close byte on the client side in tracing logs. If WebSocket works in blocking mode, it will remain in while cycle in Close() method trying to make Do0():

```
if(IsBlocking())
    while(!IsClosed() && !IsError() && socket->IsOpen())
        Do0();
```

I used upp sources from github: 993904e
Compiler: GCC 5.4.1

Subject: Re: How to close the websocket connection
Posted by [uppjj](#) on Wed, 14 Feb 2018 18:17:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello

I got the same problem, "Close" put my websocket client in infinite loop.
Server receives the "Close" message from the client but does not actually send the response, because it is in non-blocking mode.
I just do this change in uppsrc/core/WebSocket.cpp , function "out":

```
// while(IsBlocking() && socket->IsOpen() && !IsError() && out_queue.GetCount())
    while((IsBlocking() || (s[0] == CLOSE)) && socket->IsOpen() && !IsError() &&
out_queue.GetCount())
```

and got "Close" working well...

Subject: Re: How to close the websocket connection
Posted by [mirek](#) on Sat, 17 Feb 2018 11:01:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

uppjj wrote on Wed, 14 February 2018 19:17Hello

I got the same problem, "Close" put my websocket client in infinite loop.
Server receives the "Close" message from the client but does not actually send the response, because it is in non-blocking mode.
I just do this change in uppsrc/core/WebSocket.cpp , function "out":

```
// while(IsBlocking() && socket->IsOpen() && !IsError() && out_queue.GetCount())
    while((IsBlocking() || (s[0] == CLOSE)) && socket->IsOpen() && !IsError() &&
out_queue.GetCount())
```

and got "Close" working well...

Thanks. I believe this has a problem of occasionally turning nonblocking socket into blocking, as Out is used for data too.

I suggest this:

```
void WebSocket::Out(const String& s)
{
    out_queue.AddTail(s);
    while((IsBlocking() || close_sent) && socket->IsOpen() && !IsError() && out_queue.GetCount())
        Output();
}
```

Would that work for you?

Subject: Re: How to close the websocket connection

Posted by [uppij](#) on Sat, 24 Feb 2018 14:59:40 GMT

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

thanks for your reply Mirek

In my case the server no longer called do () after launching close (), I should have better read the documentation!

The problem is rather client side, see below.

I encountered several difficulties to write a client application (IoT) with Websocket class for the real world of the web, but it does not miss much:

1) the connection header is currently fixed, it is essential to be able to modify it

=> it could be something like a Vector <String> to adapt before connecting.

2) Need a public function for really close the socket. Websoket :: Close does not close the socket but sends a request to the server.

If it does not respond for some reason, the client loops indefinitely (my first problem !)

I replaced the Close by Disconnect (), and added a new Close ():

```
void Disconnect (const String & msg = Null);    // old Close (), just a message to the server
```

```
void Close () {socket-> Close ();}              // real TCP close, even is server is sleeping...
```

3) Sending Masked requests is not available. This works:

add this function in class WebSocket (Inet.h) :

```
void SendTextMasked(const String& data)          { SendRaw(MASK|FIN|TEXT, data); }
```

```

    Change SendRaw() in WebSocket.cpp :
void WebSocket::SendRaw(int hdr, const String& data)
{
    if(IsError())
        return;

    ASSERT(!close_sent);
    LLOG("Send " << data.GetCount() << " bytes, hdr: " << Format("%04X",hdr));

    // mask detect
    int LocMask = (hdr & MASK)?0x80:0;
    hdr &= 0xFF;

    //---- header construct
    // opcode
    String header;
    header.Cat(hdr);

    // Length
    int len = data.GetCount();
    if(len > 65535) {
        header.Cat(LocMask | 127);
        header.Cat(0);
        header.Cat(0);
        header.Cat(0);
        header.Cat(0);
        header.Cat(0);
        header.Cat(byte(len >> 24));
        header.Cat(byte(len >> 16));
        header.Cat(byte(len >> 8));
        header.Cat(byte(len));
    }
    else
    if(len > 125) {
        header.Cat(LocMask | 126);
        header.Cat(byte(len >> 8));
        header.Cat(byte(len));
    }
    else
        header.Cat(LocMask |(int)len);

    if (LocMask)
    {
        //add masking-key
        byte Cle[4];
        Cle[0] = Random();
        Cle[1] = Random();
        Cle[2] = Random();
        Cle[3] = Random();
    }

```

```

for(int i = 0; i < 4; i++) header.Cat(Cle[i]);

//---- send header with mask
    Out(header);

    //---- send masked data
    if(data.GetCount() != 0)
    {
        char buf[32768];
        int n = data.GetCount();
        for(int i = 0; i < n; i++)
            buf[i] = data[i] ^ (byte) Cle[i & 3];
        Out(String(buf,n));
    }
}

else
{
    //---- send header (not masked)
    Out(header);

    //--- send data (not masked)
    if(data.GetCount() != 0)
        Out(data);
}
}
hope this can help

```

Subject: Re: How to close the websocket connection

Posted by [mirek](#) on Sat, 24 Feb 2018 16:07:57 GMT

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

1. Added RequestHeader
2. It is bad to change API. Also, if simply breaking TCP connection is enough, simply destroying WebSocket should work too. Anyway, I have also added "wait_reply" parameter to Close (but that admittedly is just half of solution). Maybe the correct solution is to use non-blocking client?
3. Added with slight modification.

Please check and report.

Thanks,

Mirek

Subject: Re: How to close the websocket connection

Posted by [uppij](#) on Wed, 28 Feb 2018 14:24:32 GMT

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

Hello Mirek

Thanks for your very quick answer !

Two things to report :

1)A little bug : you forgot to add mask for short lengths (line 453):

```
(if(len > 125) {  
    header.Cat(126 | mask);  
    ...  
    else  
    header.Cat((int)len | mask);
```

2)Url connection not so easy:

RequestHeader() must be completed before using "Connect" ...but needs informations calculated by Connect (uri, host) !

Header beginning example :

```
"GET " + uri + " HTTP/1.1\r\n"
```

```
"Host: " + host + "\r\n"
```

```
...
```

in your code uri = url (not very useful) so I use uri to separate "Host" (ex: ws://demos.kaazing.com) and sub-adress (ex : /echo) , because this is (in my case) required by the remote server:

```
//--- Port  
int port = ssl ? 443 : 80;  
if(*u == ':')  
    port = ScanInt(u + 1, &u);  
  
//--- uri (JJ)  
while(*u && *u != '/' && *u != '?') u++;  
if (*u !=0) uri = u;           // only sub-adress
```

Url is the complete adress and uri is the sub-adress than can be used by GET

Solution should be to remove host, port, uri (and may be websocket-key to) calculations from Connect, into a separate function that could be used before Connect() to complete RequestHeader().

Best Regards

Jean-Jacques

Subject: Re: How to close the websocket connection

Posted by [mirek](#) on Wed, 28 Feb 2018 15:25:58 GMT

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

You are right. Can you check current trunk changes (I have tried to improve it)?

Mirek

Subject: Re: How to close the websocket connection
Posted by [Klugier](#) on Wed, 28 Feb 2018 22:27:39 GMT

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

Hello Mirek,

Why not to keep request_headers as Vector:

```
Vector<String> request_headers;
```

```
request_headers = {  
    "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",  
    "Accept-Language: cs,en-US;q=0.7,en;q=0.3",  
    ...  
};
```

// The user can add custom headers to the vector with more straight forward way. Moreover user will not be obligated to remember about /r/n.

```
auto header = web_socket.StandardHeaders();  
header.Add("Custom-header: data");
```

// Deletion is also trivial of any of the standard headers.

```
request_headers.Remove(1); // Remove "Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "I don't need it!
```

// The set method also needs to be change

```
web_socket.RequestHeaders(request_headers);
```

\r\n can be added to the lines while processing. These approach is more flexible than String implementatino. Please let me know what do you think?

Sincerely,
Klugier

Subject: Re: How to close the websocket connection

Posted by [upppj](#) on Thu, 01 Mar 2018 15:55:36 GMT

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

hello all

Mirek,

-in SenRaw you also have to add "mask" line 459 :

```
else header.Cat((int)len); => else header.Cat((int)len | mask);
```

-your improvement ... no longer allows me to use "connect". Let me explain :

the server reject : "Get ws://serveur.example.com/chat/sensor-123 HTTP/1.1" // the complete adress

but accept : "Get /chat/sensor-123 HTTP/1.1" // sub adress

if followed by : "Host: server.example.com" // site adress

it is therefore essential to have acces to uri, which in some cases may be different from url.

For the Vector, I am not sure a "Standard header" exist. User should have to delete unwanted items, which is a more complex job than to build all the Header. May be a vectormap with some predefined key "Get", "Host", "Upgrade", "Sec-WebSocket-Key" etc.. would faciliate the use with "/r/n" taking in account of course.

A good idea should be to add the Websocket Protocol specification RF6455 example in the upp Websocket documentation, to help the user build his own header:

The handshake from the client looks as follows:

```
GET /chat HTTP/1.1
```

```
Host: server.example.com
```

```
Upgrade: websocket
```

```
Connection: Upgrade
```

```
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
```

```
Origin: http://example.com
```

```
Sec-WebSocket-Protocol: chat, superchat
```

```
Sec-WebSocket-Version: 13
```

JJ

Subject: Re: How to close the websocket connection

Posted by [mirek](#) on Fri, 02 Mar 2018 14:28:55 GMT

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

mask issue fixed.

Anyway, I do not understand uri issues. That part is not changed (host is now fixed).

Other than that, I can certainly make the interface better.

Maybe follow HttpRequest interface with

Header(const char *id, String value);

Actual header would then be created as combination of these values and default values.

Mirek

Subject: Re: How to close the websocket connection

Posted by [uppjj](#) on Fri, 02 Mar 2018 23:18:34 GMT

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

Some web sites require an address like this:

"Get ws://serveur.example.com/subfolder/... HTTP/1.1" and don't care about "Host:..."

others require an address like that (and reject the above):

"Get /subfolder/... HTTP/1.1"

"Host: serveur.example.com"

I am in the second case for my project.

as WebSocket::Connect("ws://...") now always send message beginning with "Get ws://..." it is impossible to connect to second kind of site.

for the moment I use the previous version of your websocket.cpp which gives the complete control of the header, with full unfixed request_header :

```
Nvl(request_header,           // complete control
"GET " + uri + " HTTP/1.1\r\n" // rejected by some web sites
"Host: " + host + "\r\n"
```

It's not pure soft because I have to run:

- a copy of the beginning of WebSocket::Connect to separate "Host" and "subfolder" from url,
- a copy of random websocket key computing,
- then run connect() with url again (which is not used but must be correct to avoid a "return false"...))

but it does the job !

I do not see any difference between url and uri in websocket.cpp.

have a nice week end

jj

Subject: Re: How to close the websocket connection

Posted by [mirek](#) on Sat, 03 Mar 2018 09:19:51 GMT

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

OK, can you check now please? I have tried to resolve the issue with second Connect variant...

Subject: Re: How to close the websocket connection

Posted by [uppjj](#) on Sun, 04 Mar 2018 14:51:31 GMT

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

Ok thanks. This is working now with your new code, with url containing subfolder or not :

```
// adress
String Host,SubFolder;
bool ssl;
Adress_Split(Url,Host,SubFolder,ssl);
// header
WS_Client.ClearHeaders().Header("Upgrade","websocket").Header(
"Connection","Upgrade").Header("Sec-WebSocket-Version", "13"); // may be standard part of
header ?
WS_Client.Header("Sec-WebSocket-Protocol","your_protocol")...; // custom part
// connect
WS_Client.Connect(SubFolder,Host,ssl); // ok working now !
...
```

with :

```
void Adress_Split(const String& url,String& host,String& Subfolder,bool& ssl)
{ // copy of the beginning of connect(url)
  Subfolder = url;
  const char *u = url;
  ssl = memcmp(u, "wss", 3) == 0;

  //--- Host
  const char *t = u;
  while(*t && *t != '?')
    if(*t++ == '/' && *t == '/')
    {
      u = ++t;
      break;
    }
  t = u;
  while(*u && *u != ':' && *u != '/' && *u != '?')
    u++;
  host = String(t, u);

  //--- SubFolder (JJ)
```

```
while(*u && *u != '/' && *u != '?') u++;  
if (*u !=0) Subfolder = u;  
}
```
