Subject: [BUG - solved?] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Tue, 08 Jan 2013 22:36:40 GMT

View Forum Message <> Reply to Message

This sample (for latest SVN) code fails under FreeBSD

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

using namespace Upp;

CONSOLE_APP_MAIN
{
 HttpRequest::Trace(true);
 HttpRequest http("http://google.com"); //fails with almost any sites even w/o redirect
 String s = http.Execute();
 SaveFile("1", s);
}
```

It logs error:Quote:HTTP START

```
Starting status 2 'Resolving host name', url: google.com
HTTP StartConnect
HTTP AfterConnect
Starting status 6 'Sending request', url: google.com
HTTP REQUEST google.com:0
HTTP request:
GET / HTTP/1.1
URL: http://google.com/
Host: google.com
Connection: close
Accept: */*
Accept-Encoding: gzip
User-Agent: U++ HTTP request


HTTP Execute: Sending request
HTTP retry on error socket(5) / send: Socket is not connected
Starting status 1 'Start', url: google.com
HTTP Execute: Start
HTTP START
Starting status 2 'Resolving host name', url: google.com
HTTP StartConnect
HTTP AfterConnect
Starting status 6 'Sending request', url: google.com
HTTP REQUEST google.com:0
HTTP request:
GET / HTTP/1.1
URL: http://google.com/
Host: google.com
```

```
Connection: close
Accept: */*
Accept-Encoding: gzip
User-Agent: U++ HTTP request


HTTP Execute: Sending request
HTTP retry on error socket(5) / send: Socket is not connected
Starting status 1 'Start', url: google.com
HTTP Execute: Start
HTTP START
Starting status 2 'Resolving host name', url: google.com
HTTP StartConnect
HTTP AfterConnect
Starting status 6 'Sending request', url: google.com
HTTP REQUEST google.com:0
HTTP request:
GET / HTTP/1.1
URL: http://google.com/
Host: google.com
Connection: close
Accept: */*
Accept-Encoding: gzip
User-Agent: U++ HTTP request


HTTP Execute: Sending request
HTTP retry on error socket(5) / send: Socket is not connected
Starting status 1 'Start', url: google.com
HTTP Execute: Start
HTTP START
Starting status 2 'Resolving host name', url: google.com
HTTP StartConnect
HTTP AfterConnect
Starting status 6 'Sending request', url: google.com
HTTP REQUEST google.com:0
HTTP request:
GET / HTTP/1.1
URL: http://google.com/
Host: google.com
Connection: close
Accept: */*
Accept-Encoding: gzip
User-Agent: U++ HTTP request


HTTP Execute: Sending request
```

Very strange bug.
Proceeding with investigation...

P.S. Tested on defferent systems with different FreeBSD versions with the same result.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Tue, 08 Jan 2013 23:02:48 GMT
View Forum Message <> Reply to Message

It works with step-by-step mode, when breakpoints are fired after each internal iteration within HttpRequest::Execute()

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Wed, 09 Jan 2013 08:11:26 GMT
View Forum Message <> Reply to Message

It looks like the code lacks some kind of delay (really got to find out why). Still adding small delay makes it all working:

```
bool HttpRequest::Do()
{
 Sleep(10); // <-- delay added, and the code works
 int c1, c2;
 switch(phase) {
```

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by nlneilson on Wed, 09 Jan 2013 15:59:41 GMT
View Forum Message <> Reply to Message

That is similar to a socket connection with client/server.

With Win or Ubuntu with C++, Java, Python, etc. there is time involved.
Serial Port connections also require Sleep(?).

The amount of data sent or received by a Request has little effect.
It is the time required between each packet sent or received.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Thu, 10 Jan 2013 14:26:05 GMT

View Forum Message <> Reply to Message

Hello, Neil!

Frankly speaking I just did not catch your point.
We have following simple code:

```
 HttpRequest http("http://google.com");
 String s = http.Execute();
```

This code works under Windows. But fails under FreeBSD.
My point is it MUST be working under FreeBSD as well. This is how cross-platform calls work.
You agree, aren't you?

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by nlneilson on Thu, 10 Jan 2013 16:24:40 GMT
View Forum Message <> Reply to Message

Hi Paval

Yes I agree on cross-platform.

I have never tried FreeBSD.
Maybe that OS is lacking something that is necessary.

Take Android as an example which is lacking in many respects.
The Java app I work with tried porting to Android but have dropped it.
"The WW Android effort has stopped, BTW."
 http://forum.worldwindcentral.com/showthread.php?t=35031&amp ;highlight=Android

My post was just trying to show the necessity of time for any data transfer.

If Win, POSIX, MacOS work but FreeBSD does not then it could be considered a 'bug ' in U++ but trying to support ALL OSs like Android would be a real chore.

It's good you figured out what the problem was and got it working.

Neil

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Thu, 10 Jan 2013 16:52:31 GMT
View Forum Message <> Reply to Message

The problem IS NOT solved.
I wonder what U++ authors will answer.
The proposed Sleep calls are no more than dirty hack and finally the origin of such behaviour is

unknown. Besides even the hack proposed doesn't always work which means that internal cycles lack some necessary (under UNIX) delays. Investigation continues.
I'd greatly appreciate any practical theories what could it be.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Thu, 10 Jan 2013 19:33:50 GMT
View Forum Message <> Reply to Message

OK, finally managed to find a possible cause of this misfunction. In simple words, under FreeBSD, connecting socket didn't finish at the moment when we start sending data through it. Which resulted in failure with error code "Socket not connected".

So here is my solution which seems to work but may be not that good.
(Core/Socket.spp)

```
bool TcpSocket::RawConnect(addrinfo *arp)
...
//@445 +++
    if (GetErrorCode() == SOCKERR(EINPROGRESS))
     WaitWrite();
//    +++
```

If you know better solution how to detect the need and wait socket connection, please propose it.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by mirek on Thu, 24 Jan 2013 18:34:59 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 10 January 2013 14:33OK, finally managed to find a possible cause of this misfunction. In simple words, under FreeBSD, connecting socket didn't finish at the moment when we start sending data through it. Which resulted in failure with error code "Socket not connected".

So here is my solution which seems to work but may be not that good.
(Core/Socket.spp)

```
bool TcpSocket::RawConnect(addrinfo *arp)
...
//@445 +++
    if (GetErrorCode() == SOCKERR(EINPROGRESS))
     WaitWrite();
//    +++
```

If you know better solution how to detect the need and wait socket connection, please propose it.

Ha! I know that there are differences in handling this in Linux/Windows. I was not aware that BSD is going to be different as well...

Mirek

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by mirek on Thu, 24 Jan 2013 18:40:56 GMT

View Forum Message <> Reply to Message

It indeed is not that good. Anyway, perhaps it would be possible to use WIN32 handling (I think it is quite likely that Win32 modeled its non-blocking sockets model after BSD and it is Linux who diverged). Could you try that?

Mirek

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Thu, 24 Jan 2013 19:11:56 GMT

View Forum Message <> Reply to Message

Yes, I'd like to help with that, but for now couldn't find how to change sources the way you proposed. The RawConnect() function doesn't contain any ifdef WIN32.
If you tell me what to change (or just give me a clue) I'll try to.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by mirek on Thu, 24 Jan 2013 20:33:46 GMT

View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 24 January 2013 14:11Yes, I'd like to help with that, but for now couldn't find how to change sources the way you proposed. The RawConnect() function doesn't contain any ifdef WIN32.
If you tell me what to change (or just give me a clue) I'll try to.

I think that the real stuff is in "WouldBlock" method - this one is supposed to return "true" if connection is not yet established.

Mirek

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Thu, 24 Jan 2013 21:39:01 GMT

View Forum Message <> Reply to Message

After a bunch of experiments I've finally come to code which works under one of my test FreeBSD

systems I currently work in. Tomorrow I'll test it under different FreebBSD system to be absolutely sure about it.
For now, here is WouldBlock() function which seems like a solution:

```
bool TcpSocket::WouldBlock()
{
 int c = GetErrorCode();
#ifdef PLATFORM_POSIX
 #ifdef PLATFORM_BSD
 if(c == SOCKERR(ENOTCONN))
  return true;
 #endif
 return c == SOCKERR(EWOULDBLOCK) || c == SOCKERR(EAGAIN);
#endif
#ifdef PLATFORM_WIN32
 if(c == SOCKERR(ENOTCONN) && !IsNull(connection_start) && msecs(connection_start) <
20000) {
  LLOG("ENOTCONN issue");
  return true;
 }
 return c == SOCKERR(EWOULDBLOCK);
#endif
}
```

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by mirek on Fri, 25 Jan 2013 07:22:38 GMT

View Forum Message <> Reply to Message

Excellent, looks like out theory about Win sockets modeled after BSD is sound.

Anyway, in that case, we will probable have to use the whole thing, including "connection_start" - that one is supposed to detect connection timeout, otherwise it will freeze on any request with failed connection - unfortunately there does not seem to be a way how to detect this in any other way.

(You can check, IMO all you need is to do HttpRequest on non-existing host - I think it will freeze, forever waiting for connection to finish).

Mirek

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Fri, 25 Jan 2013 10:32:15 GMT

View Forum Message <> Reply to Message

It may seem srange, but HttpRequest doesn't freeze on inexistent URL. It seem to return very quickly returning empty string, which looks like correct behaviour to me.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by mirek on Fri, 25 Jan 2013 12:23:28 GMT
View Forum Message <> Reply to Message

OK, it is some time since I have added that timeout check, so perhaps the constellation of conditions for the freeze is more complex. But I do remember freezing in Win32... In any case, I think 20s timeout for connection would not hurt.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Fri, 25 Jan 2013 21:20:50 GMT
View Forum Message <> Reply to Message

Timeout check added.

```
Index: Socket.cpp
===================================================================
--- Socket.cpp (revision 5727)
+++ Socket.cpp (working copy)
@@ -268,7 +268,7 @@
  sslinfo.Clear();
  start_time = Null;
  global_timeout = Null;
-#ifdef PLATFORM_WIN32
+#if (PLATFORM_WIN32 || PLATFORM_BSD)
  connection_start = Null;
 #endif
  ssl_start = Null;
@@ -293,6 +293,9 @@
   return false;
  }
 #else
+ #if (PLATFORM_BSD)
+ connection_start = msecs();
+ #endif
  if(fcntl(socket, F_SETFL, (fcntl(socket, F_GETFL, 0) | O_NONBLOCK))) {
   SetSockError("fcntl(O_[NON]BLOCK)");
   return false;
@@ -509,7 +512,12 @@
 {
  int c = GetErrorCode();
 #ifdef PLATFORM_POSIX
- return c == SOCKERR(EWOULDBLOCK) || c == SOCKERR(EAGAIN);
+ #ifdef PLATFORM_BSD
+ DUMP(connection_start);
+ if(c == SOCKERR(ENOTCONN) && !IsNull(connection_start) && msecs(connection_start) <
20000)
+  return true;
+ #endif
+ return c == c == SOCKERR(EWOULDBLOCK) || c == SOCKERR(EAGAIN);
 #endif
 #ifdef PLATFORM_WIN32
  if(c == SOCKERR(ENOTCONN) && !IsNull(connection_start) && msecs(connection_start) <
20000) {
@@ -599,7 +607,7 @@
    return false;
   }
   if(avail > 0) {
-  #ifdef PLATFORM_WIN32
+  #if (PLATFORM_WIN32 || PLATFORM_BSD)
    connection_start = Null;
   #endif
    return true;
@@ -630,7 +638,7 @@
 { // Compute time limit for operation, based on global timeout and per-operation timeout settings
  int o = min(IsNull(global_timeout) ? INT_MAX : start_time + global_timeout,
              IsNull(timeout) ? INT_MAX : msecs() + timeout);
-#ifdef PLATFORM_WIN32
+#if (PLATFORM_WIN32 || PLATFORM_BSD)
  if(GetErrorCode() == SOCKERR(ENOTCONN) && !IsNull(connection_start))
   if(msecs(connection_start) < 20000)
    o = connection_start + 20000;
Index: Inet.h
===================================================================
--- Inet.h (revision 5727)
+++ Inet.h (working copy)
@@ -88,7 +88,7 @@

  int                  global_timeout;
  int                  start_time;
-#ifdef PLATFORM_WIN32
+#if (PLATFORM_WIN32 || PLATFORM_BSD)
  int                  connection_start;
 #endif
  int                  ssl_start;
```

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Fri, 25 Jan 2013 22:08:51 GMT
View Forum Message <> Reply to Message

Tested /not thoroughly/ the proposed changes on another FreeBSD system (different hardware, different FreeBSD version). Everything seems to be OK.

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by mirek on Sat, 26 Jan 2013 11:44:35 GMT

View Forum Message <> Reply to Message

Patch applied (with minor changes), thank you.

Mirek

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by Mindtraveller on Sat, 26 Jan 2013 16:11:36 GMT

View Forum Message <> Reply to Message

Mirek, Core/Socket.cpp @ 519 looks like an error:
return c == c == SOCKERR(EWOULDBLOCK) || c == SOCKERR(EAGAIN);

Subject: Re: [BUG] HttpRequest fails under FreeBSD
Posted by mirek on Sat, 26 Jan 2013 19:49:43 GMT

View Forum Message <> Reply to Message

Thanks.

Mirek