
Subject: [Solved]XML Rpc client will halt when server is not running

Posted by [kasome](#) on Wed, 20 Jun 2012 10:58:23 GMT

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

The following code will halt when XML RPC Server is not running.

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

using namespace Upp;

namespace Upp {
extern bool HttpRequest_Trace__;
}

int main() {
#ifdef flagDEBUG
// LogXmlRpcRequests();
// HttpRequest_Trace__ = true;
StdLogSetup( LOG_COUT | LOG_FILE );
#endif

Time serverTime;
SetDateFormat( "%4d-%02d-%02d" );
XmlRpcRequest call( "127.0.0.1:1234" );
if( call("GetServerTime") >> serverTime ) {
LOG( "Server Time =" + Upp::AsString(serverTime) );
}
else {
LOG( Upp::Format("Error: %s", call.GetError()) );
}

return 0;
}
```

I try to trace the code, the problem seems to happen in the red block. (In upp\uppsrc\Core\Socket.cpp)

```
int TcpSocket::Put(const char *s, int length)
{
LLOG("Put " << socket << ": " << length);
ASSERT(IsOpen());
if(length < 0 && s)
length = (int)strlen(s);
if(!s || length <= 0 || IsError() || IsAbort())
```

```

return 0;
done = 0;
bool peek = false;
int end_time = GetEndTime();

while(done < length) {
if(peek && !Wait(WAIT_WRITE, end_time))
return done;
peek = false;
int count = Send(s + done, length - done);
if(IsError() || timeout == 0 && count == 0 && peek)
return done;
if(count > 0)
done += count;
else
peek = true;
}

LLOG("//Put() -> " << done);
return done;
}

```

I modify the code from (In upp\uppsrc\Core\Socket.cpp)

```

bool TcpSocket::WouldBlock()
{
int c = GetErrorCode();
#ifdef PLATFORM_POSIX
return c == SOCKERR(EWOULDBLOCK) || c == SOCKERR(EAGAIN);
#endif
#ifdef PLATFORM_WIN32
return c == SOCKERR(EWOULDBLOCK) || c == SOCKERR(ENOTCONN);
#endif
}

```

to

```

bool TcpSocket::WouldBlock()
{
int c = GetErrorCode();
#ifdef PLATFORM_POSIX
return c == SOCKERR(EWOULDBLOCK) || c == SOCKERR(EAGAIN);
#endif
}

```

```
#ifdef PLATFORM_WIN32
return c == SOCKERR(EWOULDBLOCK) || c != SOCKERR(ENOTCONN);
#endif
}
```

I hope that will fix the problem.
