
Subject: Urr Data not properly returned in our case
Posted by [sinpeople](#) on Sat, 30 Jan 2021 01:44:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi folks,

I zoomed in to the case in which udp data sent from win10 sdk is not properly handled by Urr.

I did the following to isolate the problem.

A) send and receive udp data by using win10 sdk for both send and receive. (No data missing)

B) send and receive udp data by using Urr protocol call for both send and receive (No data missing)

C) send udp from win10 sdk to Urr receive (First 20 characters are missing)

I cannot tell what's the difference that caused the failure of cross-platform handling to receive UDP data from Win10 sdk. I have attached all my relevant source code here. I deleted most of those irrelevant portions and it can be compiled.

I looked into the Urr server.cpp and I found a 20 in the source code for handling the receiving data in the buffer. But I am such a newbie that I cannot understand what it is trying to do there.

Thank you very much for looking into this issue. Please advise how I can make change to correctly get data sent from win10 sdk. Though I have the win10 sdk sample code here, I prefer solve the problem at U++ side because U++ is the platform I choose to do my work.

Hope I have provided sufficient info and all source code are runnable at your side.

Best Regards

David WANG

P.S. I also tried to send data from U++ Urr and receive it by win10 sdk program. Win10 sdk program crashed immediately after I launch the sending program.

File Attachments

1) [20charMissing.PNG](#), downloaded 774 times

```

#include <CtrlLib/CtrlLib.h>
using namespace Upp;

#define LAYOUTFILE <Mta...
#include <CtrlCore/lay...
#include <Urr/Urr.h>

//thread GUI (main appl...
class MainWindow : publ...
{
public: //thread GUI
    MainWindow();

public: //thread Worker
private:
    void UDPRpcServerRecvSignalStatusThread();

MainWindow::MainWindow()
{
    CtrlLayout(*this);

    Thread udprpcRecvSigStatus;
    udprpcRecvSigStatus.Run(callback(this, &MainWindow::UDPRpcServerRecvSignalStatusThread));
}

void MainWindow::UDPRpcServerRecvSignalStatusThread()
{
    UrrServer urr;

    urr.Create(40001); //7593<==
    //Cout() << "URR Ping server\n";
    //while(false)
    for(;;)
    {
        UrrRequest r;
        if(urr.Accept(r))
        {
            String signalStatus = ~r;
            log.Set(log.Get()+signalStatus);
        }
        //Sleep(200);
    }
}

GUI_APP_MAIN
{
    MainWindow().Sizeable().Run();
}

```

Urr Receive Window

5,2,4,1,5,2,4,1,5,2,4,1,5,2,4

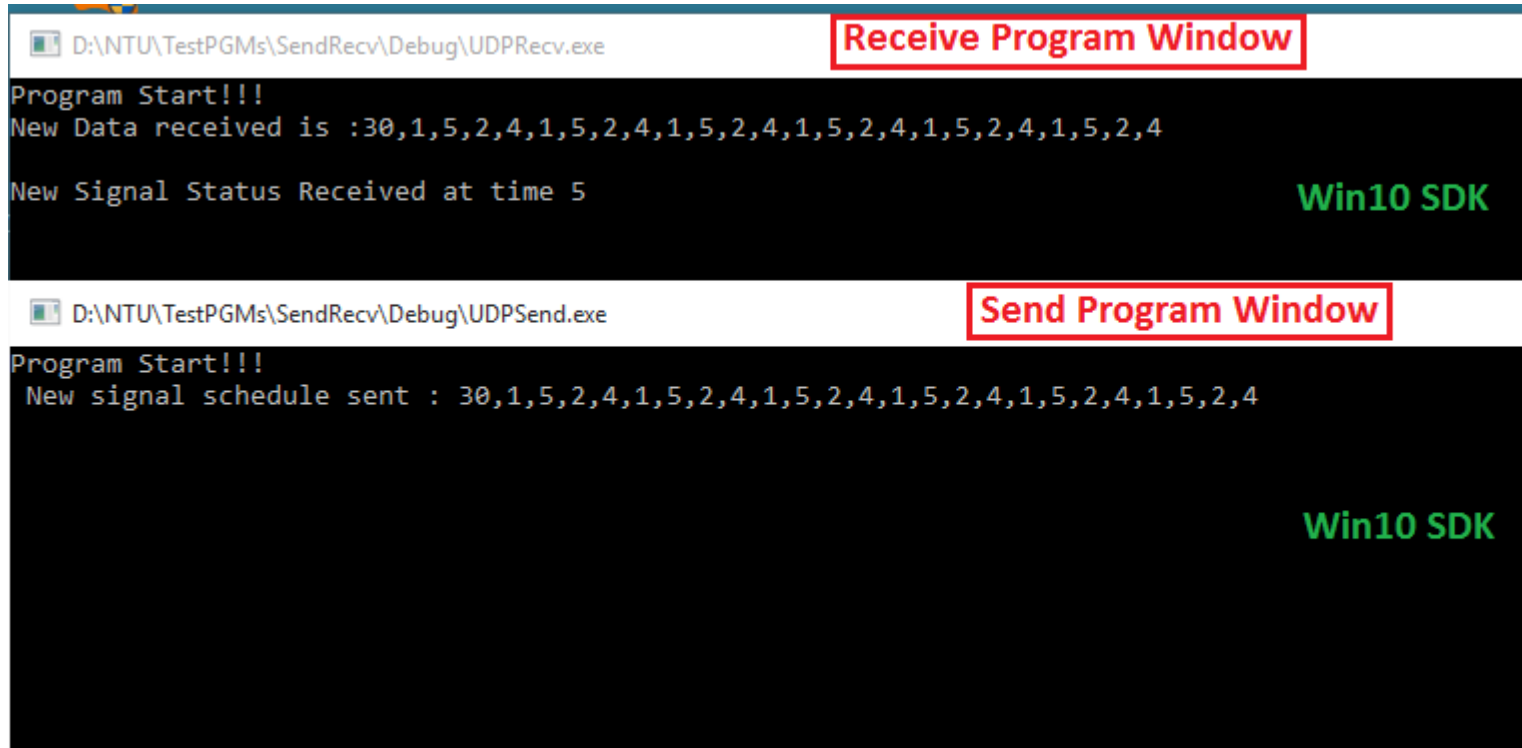
U++ Urr

D:\NTU\TestPGMs\SendRe

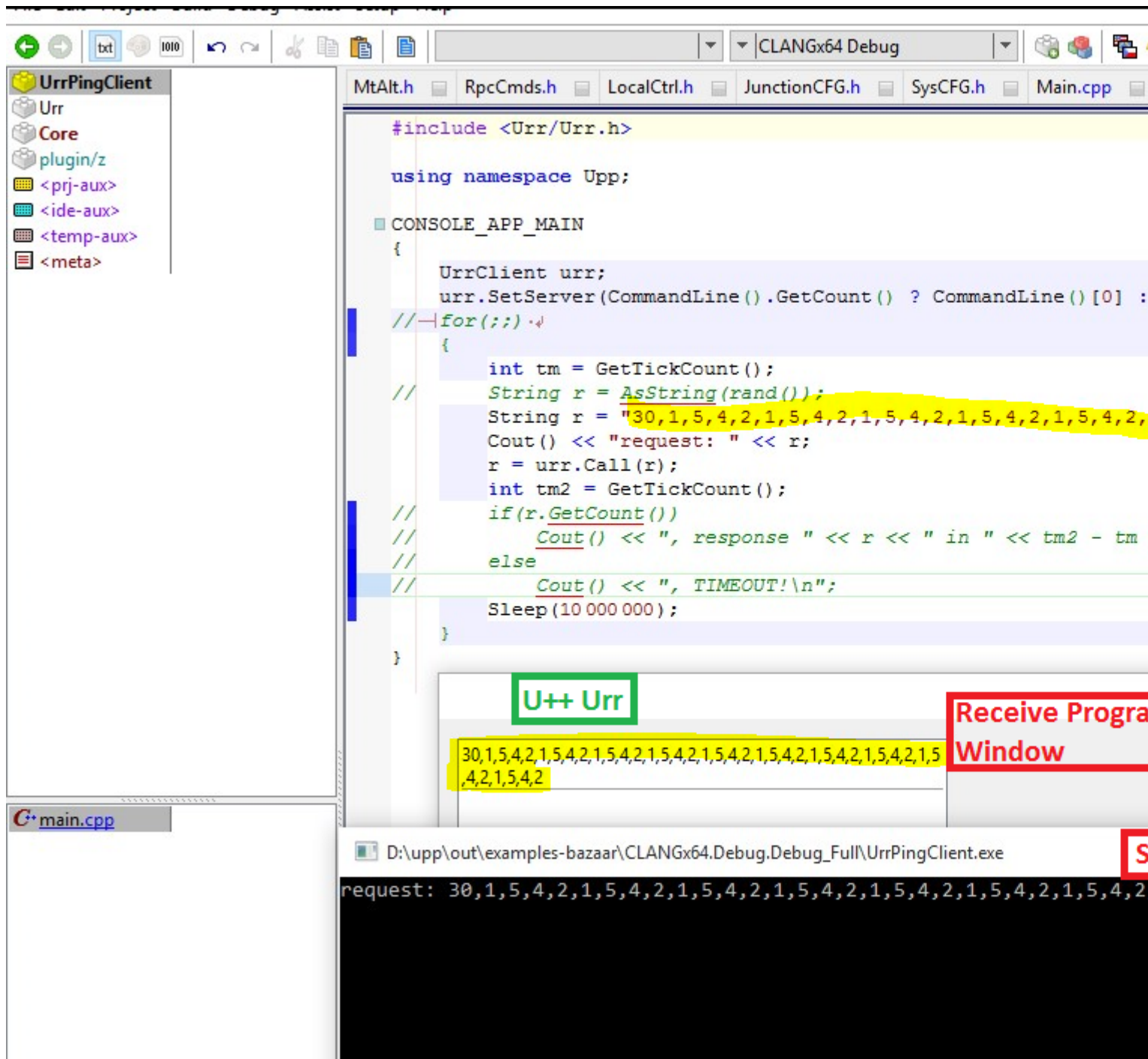
Program Start!!!
New signal schedule

Urr Receive Source Code

2) [SendRecv.PNG](#), downloaded 770 times



3) [UrrPingOK.PNG](#), downloaded 753 times



- 4) [w10_SendRecv.zip](#), downloaded 296 times
- 5) [UrrRecvTestPGM.zip](#), downloaded 310 times

Subject: Re: Urr Data not properly returned in our case
Posted by [mirek](#) on Sun, 07 Feb 2021 09:22:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

You misunderstand the URR.

URR IS NOT UDP ENCAPSULATION LIBRARY!

Win10 SDK is UDP protocol example. UDP protocol itself has some limitations.

URR is (experimental) protocol based on UDP that adds some bookkeeping data to UDP in an attempt to "fix" these limitation. That is why there are data missing when you receive UDP datagram - URR interprets the beginning as header.

Frankly, it is low importance project, that is why it is not in "canonical" U++ (uppsrc). I have uploaded it into bazaar because I have it developed for some project that was later abandoned, it works, it is tested, but has limited utility, unless you hit the same usage scenario (single server that has to process thousands of requests per second - HTTP is too slow for that).

If you desperately need UDP in U++, well, it is really trivial and you can cut those lines of code from URR.

Mirek
