

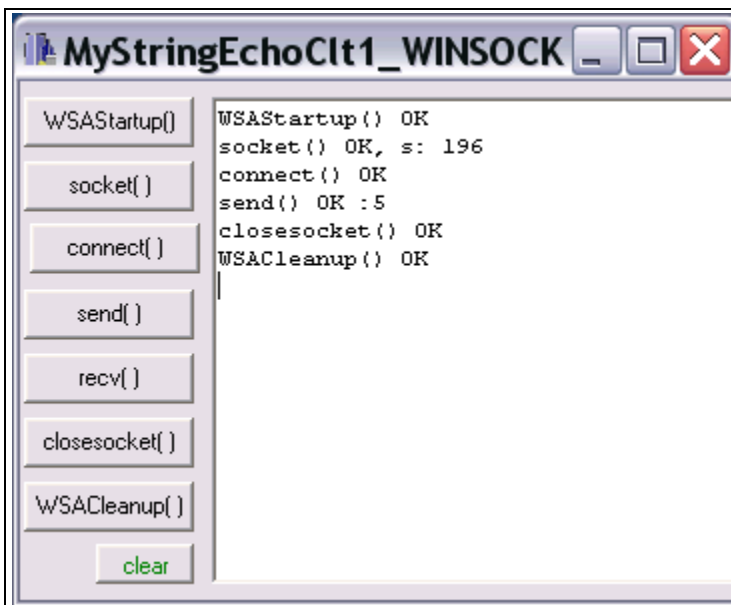
## 1.1 Ziel des Projektes

- Es soll ein String-Echo-Clt auf WINSOCK aufgesetzt werden.
- Der Client sendet an Server einen Text, der in Großbuchstaben gewandelt zurück gesendet wird.
- Der Server arbeite lokal auf Port 6666.
- Seine IP-Adresse lautet "127.0.0.1" oder der Name "localhost"

## 1.2 Voraussetzungen

Laden Sie zuerst den StringEchoServer herunter und starten diesen lokal!

## 1.3 Realisierung des Projektes MyStringEchoClt1\_WINSOCK



- Erzeugen Sie ein neues Projekt "MyStringEchoClt1\_WINSOCK" im gleichnamigen Order.
- Editieren Sie die Oberfläche entsprechend der nebenstehenden Abbildung.
- Erzeugen Sie Schritt für Schritt den Programmcode, entsprechend dem Beispiel-Code.

```
#include <vcl.h>
#include <winsock.h> //erforderlich
#include <stdio.h> //erforderlich z.B. für putchar()
#include <string>
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int s; //socket descriptor
int len; //für send() und recv()
char buf[4095]; //für send() und recv()
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{ }
//-----
void __fastcall TForm1::WSAStartup_Click(TObject *Sender)
{ //==(1)Am API anmelden
  WSADATA wsad;
  if (WSAStartup(MAKEWORD(2,0),&wsad) ==0) //MAKEWORD(maj,min)
    RichEdit1->Lines->Add("WSAStartup() OK");
  else
```

```

    RichEdit1->Lines->Add("WSAStartup() ERR");
}
void __fastcall TForm1::socket_Click(TObject *Sender)
{
    //==(2)Socket errichten
    s = socket(AF_INET,SOCK_STREAM,0);
    if (s!=0) RichEdit1->Lines->Add("socket() OK, s: "+IntToStr(s));
    else RichEdit1->Lines->Add("socket() ERR");
}
//-----
void __fastcall TForm1::connect_Click(TObject *Sender)
{
    //==(3)Adresse in Struktur sockaddr_in eintragen
    sockaddr_in sAddr;
    sAddr.sin_family=AF_INET;
    sAddr.sin_port=htons(6666);
    sAddr.sin_addr.s_addr=inet_addr("127.0.0.1");
    /*
    sAddr.sin_addr.S_un.S_un_b.s_b1=127; sAddr.sin_addr.S_un.S_un_b.s_b2=0;
    sAddr.sin_addr.S_un.S_un_b.s_b3=0; sAddr.sin_addr.S_un.S_un_b.s_b4=1;
    sAddr.sin_addr.S_un.S_un_w.s_w1=MAKEWORD(127,1);
    sAddr.sin_addr.S_un.S_addr=MAKELONG(MAKEWORD(127,2),MAKEWORD(0,1));
    sAddr.sin_addr.s_addr=htonl(0xC0A80001); */
    //==(4)Verbindung herstellen, das Ziel geht aus sAddr hervor
    if (connect (s,(sockaddr*)&sAddr,sizeof(sockaddr))==0)
        RichEdit1->Lines->Add("connect() OK");
    else
        RichEdit1->Lines->Add("connect() ERR: "+IntToStr(WSAGetLastError()));
}
//-----
void __fastcall TForm1::recv_Click(TObject *Sender)
{
    //==(5)Daten aus dem Socket lesen
    len = recv(s, buf, sizeof(buf),0);
    if ((len!=0)&& (len!=SOCKET_ERROR))
    {
        RichEdit1->Lines->Add("recv() OK :"+IntToStr(len));
        buf[len]=0;
        RichEdit1->Lines->Add(buf);
    }
}
//-----
void __fastcall TForm1::send_Click(TObject *Sender)
{
    //==(5)Daten in den Socket schreiben
    char *buf="hallo"+0;
    len =send (s,buf,strlen(buf),0);
    if (len!=SOCKET_ERROR)
        RichEdit1->Lines->Add("send() OK :"+IntToStr(strlen(buf)));
    else
        RichEdit1->Lines->Add("connect() ERR: "+IntToStr(WSAGetLastError()));
}
//-----
void __fastcall TForm1::closesocket_Click(TObject *Sender)
{
    //==(6)Verbindung abbauen
    if (closesocket(s)!=SOCKET_ERROR) RichEdit1->Lines->Add("closesocket() OK");
    else
        RichEdit1->Lines->Add("closesocket() ERR: "+IntToStr(WSAGetLastError()));
}
//-----
void __fastcall TForm1::WSACleanup_Click(TObject *Sender)
{
    //==(7)Am API abmelden
    WSACleanup();
    RichEdit1->Lines->Add("WSACleanup() OK");
}
//-----
void __fastcall TForm1::clear_Click(TObject *Sender)
{
    RichEdit1->Clear();
}

```