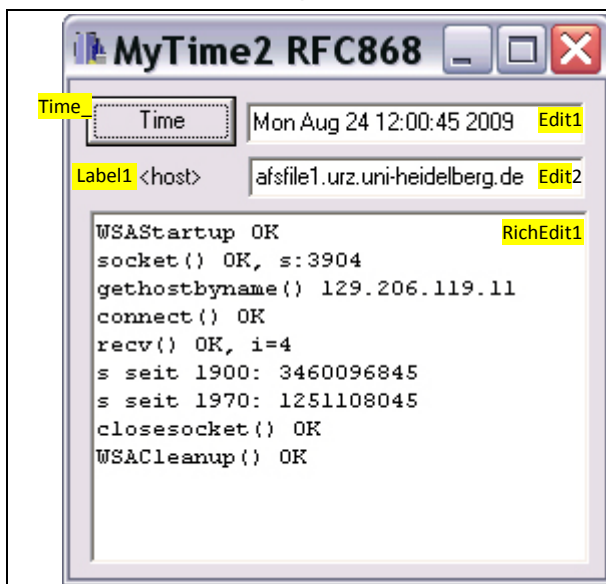


1.1 Ziel des Projektes

- Im Projekt **MyTime2_RFC868_WINSOCK** lag die IP-Adresse (141.55.192.51) des Time-Servers vor.
- Hier soll nur der Name des Servers bekannt sein:
z.B. **afsfiler1.urz.uni-heidelberg.de** oder z.B **ntp.htwm.de**
- Hier muss also zuerst die Auflösung des Rechnernamens auf seine IP-Adresse erfolgen, mittels:
 - Hilfsfunktion **gethostbyname ()**
 - und der Struktur **hostent**
- Alles andere wie beim MyDate1.

1.2 Realisierung des Projektes



- 📌 Erzeugen Sie ein neues Projekt “**MyTime2_RFC868_WINSOCK_STREAM**“ im gleichnamigen Order.
- 📌 Editieren Sie die Oberfläche entsprechend der nebenstehenden Abbildung.
- 📌 Erzeugen Sie Schritt für Schritt den Programmcode, entsprechend dem Beispiel-Code.
- 📌 Lösen Sie zuerst den Namen auf die Adresse auf.

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include <winsock.h> //erforderlich  
#include <time.h> //erforderlich  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
: TForm(Owner)  
{  
}  
//-----  
  
void __fastcall TForm1::Time_Click(TObject *Sender)  
{  
    RichEdit1->Clear();  
    //===(1) Am API anmelden; Funktion: WSAStartup(); Variable: WSADATA wsaData==  
    WSADATA wsaData;  
    if (WSAStartup(0x101, &wsaData))  
        {RichEdit1->Lines->Add("WSAStartup ERR"); goto ende;}  
    else
```

```

RichEdit1->Lines->Add("WSAStartup OK");

//===(2) Socket errichten; Funktion: socket(); Merke: s =====
int s;//Socket-Descriptor
s=socket(AF_INET,SOCK_STREAM,0);
if (s==0)
{RichEdit1->Lines->Add("socket() ERR"); goto ende;}
else
RichEdit1->Lines->Add("socket() OK, s:"+IntToStr(s));

//===(3a) IP-Adresse aus Name; Funktion: gethostbyname();struktur: hostent===
struct hostent *host;//Zeiger host auf Struktur vom Typ hostent
host=gethostbyname(Edit2->Text.c_str());
if (host)
{
RichEdit1->Lines->Add("gethostbyname() "
+(AnsiString)(u_char)host->h_addr[0]+". "
+(AnsiString)(u_char)host->h_addr[1]+". "
+(AnsiString)(u_char)host->h_addr[2]+". "
+(AnsiString)(u_char)host->h_addr[3]
);
}
else
{RichEdit1->Lines->Add("gethostbyname() ERR"); goto ende;}

//===(3b) Hostadresse (IP, Port) eintragen; Struktur: SOCKADDR_IN =====
SOCKADDR_IN sAddr;
sAddr.sin_family=AF_INET;
sAddr.sin_port=htons(37);
//IP-Adresse aus h_addr ermitteln:
//(Cardinal*)host->h_addr: fasse host->h_addr als Zeiger auf Cardinal auf
sAddr.sin_addr.S_un.S_addr*((Cardinal*)host->h_addr);

//===(4) Verbindung herstellen;Funktion: connect() =====
if (connect(s,(struct sockaddr *) &sAddr,sizeof(sAddr))!=0)
{RichEdit1->Lines->Add("connect() ERR"); goto ende;}
else
RichEdit1->Lines->Add("connect() OK");

//===(5) Daten aus Socket lesen;Funktion: recv(); ===
int i; time_t t;
i=recv(s, (char *) &t,4,0);
if (i!=4)
{RichEdit1->Lines->Add("recv() ERR"); goto ende;}
else
{RichEdit1->Lines->Add("recv() OK, i="+IntToStr(i));
RichEdit1->Lines->Add("s seit 1900: "+(AnsiString)ntohl(t) );
t=ntohl(t)-2208988800;
RichEdit1->Lines->Add("s seit 1970: "+(AnsiString)t);
Edit1->Text=((AnsiString)asctime(localtime(&t))).SubString(0,24);
}

//===(6) Verbindung beenden; Funktion: closesocket(); =====
if (closesocket(s)!=0)
{RichEdit1->Lines->Add("closesocket() ERR"); goto ende;}
else
RichEdit1->Lines->Add("closesocket() OK");

//===(7) Am API abmelden; nur bei WINSOCK;Funktion: WSACleanup(); =====
if (WSACleanup()!=0)
{RichEdit1->Lines->Add("WSACleanup() ERR"); goto ende;}
else
RichEdit1->Lines->Add("WSACleanup() OK");
ende:
}

```