

1.1 Ziel des Projektes

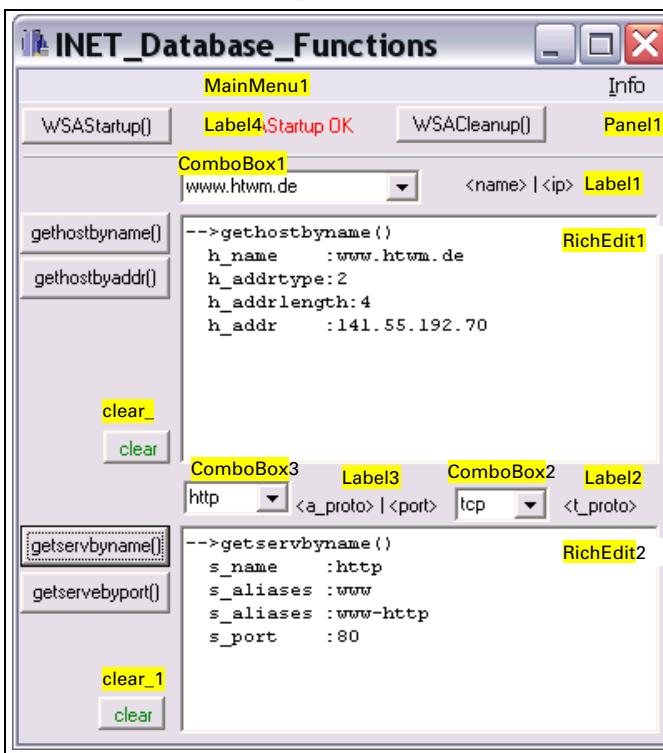
Es sollen die Datenbasisfunktionen und die entsprechenden Datenstrukturen angewendet werden.

- gethostbyname(), nutzend hostent, zur Ermittlung der IP-Adresse bei gegebenem Hostnamen
- gethostbyaddr(), nutzend hostent, zur Ermittlung des Hostnamens bei gegebener IP-Adresse.
- getservbyname(), nutzend servent, zur Ermittlung der Portnummer bei gegebenem Dienstname.
- getservbyport(), nutzend servent, zur Ermittlung des Dienstnamens bei gegebener Portnummer.

1.2 Ausprobieren und dann erst selber implementieren

Downloaden Sie zuerst INET_Database_Functions.EXE herunter und testen Sie die Funktionen. Beachten Sie, dass sie zuerst WSAShutdown() aufrufen müssen.

1.3 Realisierung des Projektes INET_Database_Functions



Erzeugen Sie ein neues Projekt "INET_Database" im gleichnamigen Order.

Erzeugen Sie die nebenstehende Oberfläche:

- TMainMenu: MainMenu1
- TPanel: Panel1 mit:
 - TButton: WSAShutdown_1, WSACleanup_1
 - TLabel: Label4
- TButton getservbyname_, getservbyport_, getservbyname_, getservbyport
- TComboBox: ComboBox1 für <name>|<ip>, ComboBox2 für <tproto>, ComboBox3 für <a_proto|port>.
- TRichEdit: RichEdit1, RichEdit2
- TBitBtn : clear_, clear_1
- TLabel: Label1, Label2, Label3,

Erzeugen Sie Schritt für Schritt den Programmcode, entsprechend dem nachfolgenden Beispiel-Code.

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include <winsock2.h>  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}
```

¹ Die Underline-Zeichen sollten Sie nicht weglassen! Komponenten dürfen nicht die Bezeichner von Funktionen haben! Das nimmt Ihnen der Compiler sehr übel und gibt Ihnen: [C++ Fehler] Unit1.cpp(37): E2314 Aufruf einer Nicht-Funktion

```

//-----
void __fastcall TForm1::WSAStartup_Click(TObject *Sender)
{
    WSADATA wsad;
    if (WSAStartup(MAKEWORD(2,0), &wsad) != 0) Label4->Caption="WSAStartup ERR";
    else Label4->Caption="WSAStartup OK";
}
//-----
void __fastcall TForm1::WSACleanup_Click(TObject *Sender)
{
    if (WSACleanup() != SOCKET_ERROR) Label4->Caption="nicht angemeldet";
    else Label4->Caption="WSACleanup ERR";
}
//-----
void __fastcall TForm1::gethostbyname_Click(TObject *Sender)
{
    struct hostent *host;
    struct sockaddr_in xAddr;
    RichEdit1->Lines->Add("-->gethostbyname()");
    //direkt: host=gethostbyname("www.google.de"); ODER aus Komponente
    if (host = gethostbyname(ComboBox1->Text.c_str()))
    {
        RichEdit1->Lines->Add(" h_name      :"+(AnsiString)host->h_name);
        //h_aliases auslesen
        while(*(host->h_aliases))
        {
            RichEdit1->Lines->Add(" h_alias    :"+(AnsiString)*host->h_aliases);
            host->h_aliases++;
        } //--endwhile
        RichEdit1->Lines->Add(" h_addrtype:"+IntToStr(host->h_addrtype));
        RichEdit1->Lines->Add(" h_addrlen:"+IntToStr(host->h_length));
        //IP-Adressen auslesen
        while(*(host->h_addr_list))
        {
            xAddr.sin_addr=*(struct in_addr*) *host->h_addr_list;
            RichEdit1->Lines->Add(" h_addr      :"+(AnsiString)inet_ntoa(xAddr.sin_addr));
            host->h_addr_list++;
        } //--endwhile
    } //--endif
    else RichEdit1->Lines->Add("Adressenaufloesung nicht moeglich!");
}
//-----
void __fastcall TForm1::clear_Click(TObject *Sender)
{
    RichEdit1->Clear();
}
//-----
void __fastcall TForm1::gethostbyaddr_Click(TObject *Sender)
{
    struct in_addr xAddr, aAddr;
    struct hostent *host;
    RichEdit1->Lines->Add("-->gethostbyaddr()");
    //xAddr.S_un.S_addr=inet_addr("141.55.192.70");
    xAddr.S_un.S_addr=inet_addr(ComboBox1->Text.c_str());
    host=gethostbyaddr((char*)&xAddr, sizeof(xAddr), AF_INET);
    if(host)
    {
        RichEdit1->Lines->Add(" h_name      :"+(AnsiString)host->h_name);
        //h_aliases auslesen
        while(*(host->h_aliases))
        {
            RichEdit1->Lines->Add(" h_alias    :"+(AnsiString)*host->h_aliases);
            host->h_aliases++;
        }
    }
}

```

```

}
RichEdit1->Lines->Add(" h_addrtype:"+IntToStr(host->h_addrtype));
RichEdit1->Lines->Add(" h_addrtype:"+IntToStr(host->h_length));
//IP-Adresse anzeigen
RichEdit1->Lines->Add(" h_addr      : "
                    +IntToStr((u_char)host->h_addr[0])+"."
                    +IntToStr((u_char)host->h_addr[1])+"."
                    +IntToStr((u_char)host->h_addr[2])+"."
                    +IntToStr((u_char)host->h_addr[3]) );
}
}
else RichEdit1->Lines->Add("Adressenaufloesung nicht moeglich!");
}
}
//-----
void __fastcall TForm1::getservbyname_Click(TObject *Sender)
{
    struct servent *serv;
    RichEdit2->Lines->Add("-->getservbyname()");
    //serv=getservbyname("http","tcp");
    if (serv=getservbyname(ComboBox3->Text.c_str(),ComboBox2->Text.c_str()))
    {
        RichEdit2->Lines->Add(" s_name      :"+(AnsiString) serv->s_name);
        while(*(serv->s_aliases))
        {
            RichEdit2->Lines->Add(" s_aliases :"+(AnsiString)*(serv->s_aliases));
            (serv->s_aliases++);
        }
        RichEdit2->Lines->Add(" s_port      :"+IntToStr(ntohs(serv->s_port)));
    }
    else RichEdit2->Lines->Add(" Aufloesung nicht moeglich!");
}
}
//-----
void __fastcall TForm1::getservebyport_Click(TObject *Sender)
{
    struct servent *serv;
    RichEdit2->Lines->Add("-->getservebyport()");
    if (serv=getservebyport(htons(ComboBox3->Text.ToInt()),ComboBox2->Text.c_str()))
    {
        //Aufruf mit direkten Parametern: serv=getservebyport(htons(80),"tcp");

        RichEdit2->Lines->Add(" s_name      :"+(AnsiString) serv->s_name);
        while(*(serv->s_aliases))
        {
            RichEdit2->Lines->Add(" s_aliases :"+(AnsiString)*(serv->s_aliases));
            (serv->s_aliases++);
        }
        }
        RichEdit2->Lines->Add(" s_port      :"+IntToStr(ntohs(serv->s_port)));
        RichEdit2->Lines->Add(" s_proto     :"+(AnsiString) serv->s_proto);
        }
        }
        else RichEdit2->Lines->Add(" Aufloesung nicht moeglich!");
    }
}
//-----
void __fastcall TForm1::clear_1Click(TObject *Sender)
{
    RichEdit2->Clear();
}
}
//-----
void __fastcall TForm1::HilfelClick(TObject *Sender)
{
    if ( Application->MessageBox("Prof. L. Winkler, Hochschule
Mittweida\r\nErstprogrammierung:\t15.02.2006\r\nAktuelle Version vom:\t15.05.2006",
"Info", MB_OK) != IDOK);
}
}
//-----

```