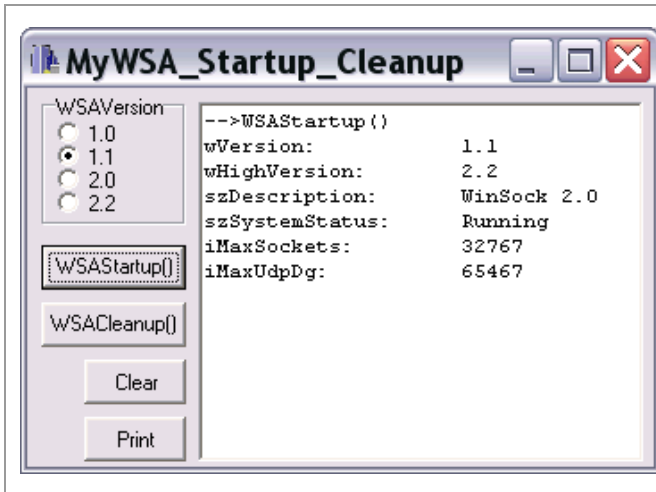


1.1 Ziel des Projektes



- Im Gegensatz zum BSD-API, muss man sich am Windows Socket API an- und abmelden.
- Die Funktionen heißen:
 - o WSAStartup()
 - o WSACleanup()
- Bei der Anmeldung gibt das API seine Eigenschaften in **struct WSADATA** zurück.
- Daraus gehen die unterstützten Versionen, die Anzahl max. Sockets usw. hervor. Einige Eigenschaften sollen angezeigt werden und druckbar sein.

1.2 Grundlagen

Verschaffen Sie sich einen Überblick:

- (1) Folie: Allgemeiner Ablauf einer WSA-Nutzung.
- (2) Folie: Struktur **struct WSADATA**
- (3) Folien: Funktion **WSAStartup()**, Funktion **WSACleanup()**.
- (4) Downloaden Sie **MyWSA_Startup_Cleanup.EXE** von www.telecom.hs-mittweida.de → Lehre Direktstudium → Socketprogrammierung – Beispiele und testen Sie die Funktionalität.

1.3 Realisierung des Projektes MyCanvas

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

Erzeugen Sie die oben gezeigte Oberfläche mittels folgender Klassen:

Klasse	Objektname	Eigenschaften/Methoden
TForm1	Form1	Caption: MyWSA...
TRadioGroup	WSAVersion	Caption: WSAVersion Columns: 1 Items: 1.0, 1.1, ... ItemIndex: 0
TButton	WSAStartup_	Caption: WSAStartup()
	WSACleanup_	Caption: WSACleanup()
	Clear_	Caption: Clear
	Print_	Caption: Print
TRichEdit	RichEdit1	
TPrintDialog	PrintDialog1	

Erzeugen Sie den Programmcode:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include <winsock2.h>  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
WSADATA wsAD;
```

```

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::WSAStartup_Click(TObject *Sender)
{
    int retWert;
    RichEdit1->Lines->Add("-->WSAStartup()");
    switch (WSAVersion->ItemIndex)
    {
        case 0: retWert=WSAStartup(MAKEWORD(1,0),&wsaD);break;
        case 1: retWert=WSAStartup(MAKEWORD(1,1),&wsaD);break;
        case 2: retWert=WSAStartup(MAKEWORD(2,0),&wsaD);break;
        case 3: retWert=WSAStartup(MAKEWORD(2,2),&wsaD);break;
    }
    if (retWert==0)
    {
        RichEdit1->Lines->Add("wVersion:\t\t"+IntToStr(wsaD.wVersion&0x00FF )
            +". "+IntToStr((wsaD.wVersion & 0xFF00)/256));
        RichEdit1->Lines->Add("wHighVersion:\t\t"+IntToStr(wsaD.wHighVersion&0x00FF )
            +". "+IntToStr((wsaD.wHighVersion & 0xFF00)/256));
        RichEdit1->Lines->Add("szDescription:\t"+(AnsiString)wsaD.szDescription);
        RichEdit1->Lines->Add("szSystemStatus:\t"+(AnsiString)wsaD.szSystemStatus);
        RichEdit1->Lines->Add("iMaxSockets:\t\t"+(AnsiString)wsaD.iMaxSockets);
        RichEdit1->Lines->Add("iMaxUdpDg:\t\t"+(AnsiString)wsaD.iMaxUdpDg);
        }//endif
    }
//-----

void __fastcall TForm1::Clear_Click(TObject *Sender)
{
    RichEdit1->Clear();
}
//-----

void __fastcall TForm1::Print_Click(TObject *Sender)
{
    if (PrintDialog1->Execute()) { RichEdit1->Print("WSAData");}
}
//-----

void __fastcall TForm1::WSACleanup_Click(TObject *Sender)
{
    RichEdit1->Lines->Add("-->WSACleanup()");
    WSACleanup();
}
//-----

```