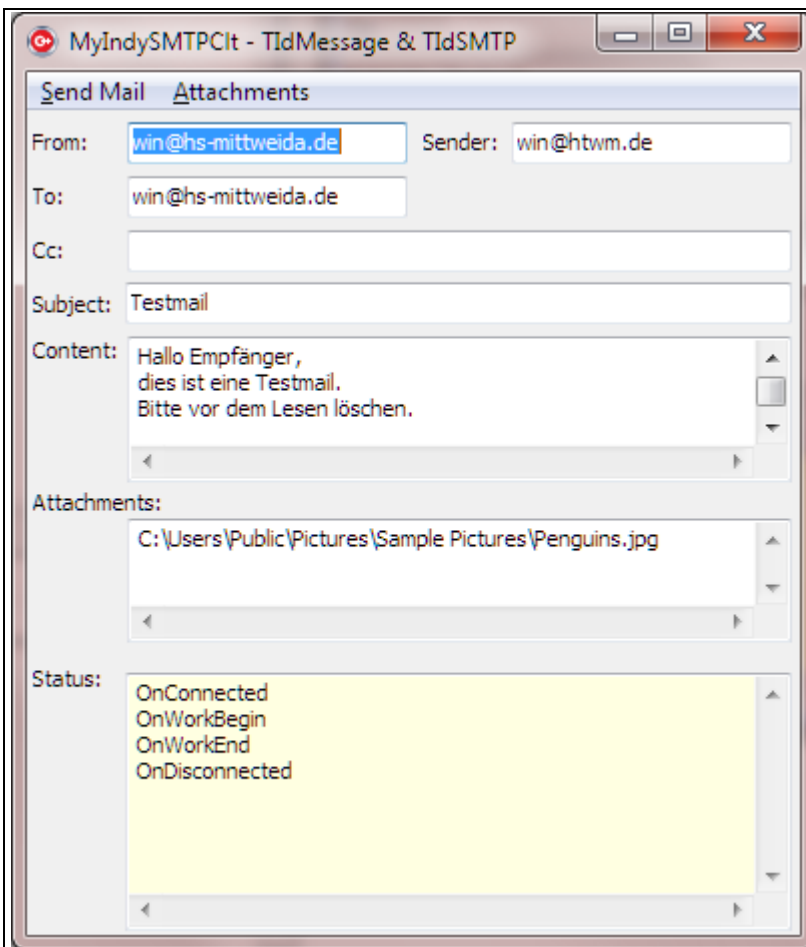


1 Ziel des Projektes

- Verwendung der Indy-Komponenten: TIdSMTP, TIdMessage
- Indy steht für „Internet direct“ → <http://www.indyproject.org/>
- **TIdSMTP** ist ein Socket mittels dem eine Verbindung zum Mail-Server aufgebaut, die Message gesendet und die Verbindung abgebaut werden kann.
- **TIdMessage** ist die Message, die bezüglich aller Eigenschaften zusammengebaut werden kann, einschließlich beliebiger Anhänge.
- Die Klassen werden nicht extra beschrieben, da der Quellcode dokumentiert wurde.

2 Realisierung des Projektes MyIndySMTPClient



- Erzeugen Sie ein neues Projekt "MyIndySMTPClient" im gleichnamigen Order.
- Editieren Sie die Oberfläche entsprechend der nebenstehenden Abbildung.
- Erzeugen Sie Schritt für Schritt den Programmcode, entsprechend dem Beispiel-Code. Verwenden Sie bitte folgende Klassen und Objektnamen:
 - o TMainMenu: MainMenu1
 - o TEdit: Edit1 .. Edit5
 - o TMemo: Memo1 .. Memo3
 - o TLabel: Label1 .. Label9
 - o TOpenDialog: OpenDialog1
 - o TIdMessage: IdMessage1
 - o TIdSMTP: IdSMTP1

//==notwendiger Programmtext==

```
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "IdAttachmentFile.hpp" //nur ab Indy10
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
```

```

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

//-----
void __fastcall TForm1::SendMailClick(TObject *Sender)
{
    Memo3->Clear();
    //==Nutzung von IdMessage, um Adressen und Inhalt der Mail zu erzeugen==

    //--Angaben zum Absender verteilt auf Properties "Name", "Address"-----
    //IdMessage1->From->Name="Prof. L. Winkler";
    IdMessage1->From->Address=Edit1->Text;
    //--oder Angaben zum Absender zusammengefasst in Property "Text"-----
    //IdMessage1->From->Text="Prof. L. Winkler win@htwm.de";

    //--Angaben zum Sender (z.B. Sekretär) in Properties "Name", "Address"-----
    //IdMessage1->Sender->Name="Frau Müller";
    //IdMessage1->Sender->Address="mueller@htwm.de";
    //--oder Angaben zum Sender zusammengefasst in Property "Text"-----
    // Mail von Sekretär im Auftrag von Chef
    if (Edit2->Text!="") { IdMessage1->Sender->Text=Edit2->Text; }

    //--Empfängeradressen, müssen durch Komma getrennt sein-----
    //--alle Empfänger erscheinen im Header "To:"
    IdMessage1->Recipients->EMailAddresses=Edit3->Text;

    //--Carbon Copy: erhalten auch Mail und erscheinen in "To:" -----
    if (Edit4->Text!=""){ IdMessage1->CCList->EMailAddresses=Edit4->Text; }

    //--Betreff
    IdMessage1->Subject=Edit5->Text;

    //--es gibt 5 Priorities: mpHighest, mpHigh, mpNormal, mpLow, mpLowest
    IdMessage1->Priority=mpNormal;

    //--Die eigentliche Nachricht
    IdMessage1->Body=Memo1->Lines;

    //--Verbindung herstellen, Nachricht senden, Verbindung beenden
    IdSMTP1->Connect();
    IdSMTP1->Send(IdMessage1);
    IdSMTP1->Disconnect();
}

//-----
void __fastcall TForm1::AttachmentsClick(TObject *Sender)
{
    //--Anhänge zur Mail hinzufügen
    if (OpenDialog1->Execute()) {
        Memo2->Lines->Add(OpenDialog1->FileName);
    }
    //->Indy10
    IdMessage1->MessageParts->Add();
    TIdAttachmentFile *AttachmentFile =
        new TIdAttachmentFile(IdMessage1->MessageParts, OpenDialog1->FileName);
    //->Indy 9: TIdAttachment(IdMessage1->MessageParts,OpenDialog1->FileName);
}

//== nicht notwendiger Programmtext

```

```

//-----
void __fastcall TForm1::IdSMTP1Connected(TObject *Sender)
{
    Memo3->Lines->Add("OnConnected");
}
//-----

void __fastcall TForm1::IdSMTP1Disconnected(TObject *Sender)
{
    Memo3->Lines->Add("OnDisconnected");
}

//-----
void __fastcall TForm1::IdSMTP1WorkBegin(TObject *ASender, TWorkMode AWorkMode,
__int64 AWorkCountMax)
{
    Memo3->Lines->Add("OnWorkBegin");
}

//-----
void __fastcall TForm1::IdSMTP1WorkEnd(TObject *ASender, TWorkMode AWorkMode)
{
    Memo3->Lines->Add("OnWorkEnd");
}
//-----

```