

计算机网络实验—基于SMTP发送邮件



班 级： 电气 16 班

姓 名： 马 丽 军

学 号： 01041172

完成时间： 2004 年 5 月 26 日

指导教师： 薛 涛

日期： 2004 年 5 月 27 日

目 录

一、实验任务·····	3
二、实验环境·····	3
三、设计思路·····	4
四、实验过程·····	5
五、实验故障与排除·····	15
六、个人总结·····	16
附录·····	16

一、实验任务

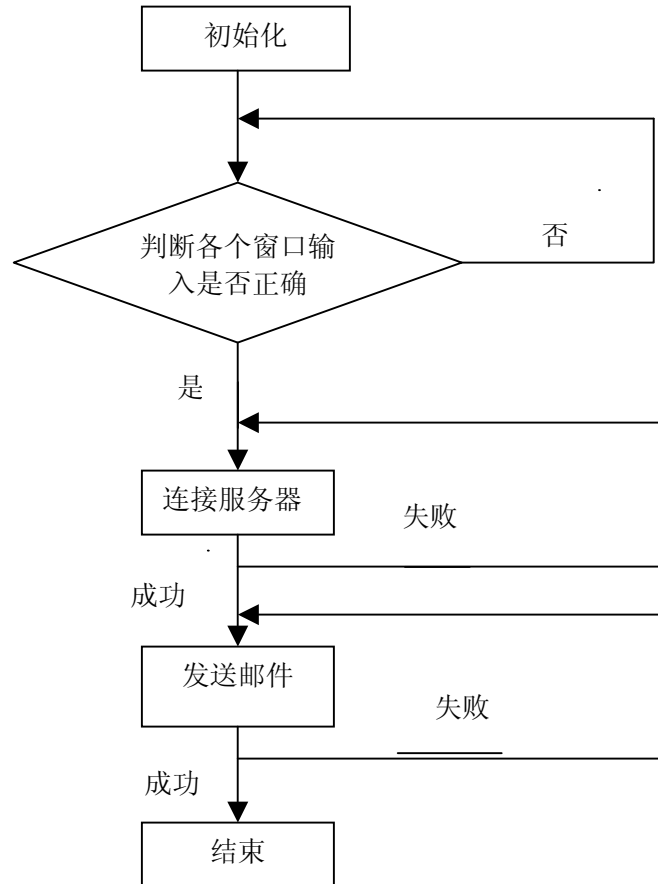
现在电子邮件的应用越来越广泛，它逐渐被人们所接受使用，因为它具有方便快捷，经济的优点。而发送邮件的形式有借助与IE或者OutLook，而本人经常IE，不过这还要先打开IE浏览器，再进入邮箱进行操作，很麻烦，所以在做网络实验的过程中本人决定开发一个电子邮件发送程序，它可以借助SMTP服务器来发送邮件。因此，我制作了一个便于发送邮件的基于smtp的程序如图：



二、实验环境

1. Pentium 166/128M 内存
2. Windows 95/98/2000/Me/Xp 操作系统
3. 基于 SMTP 协议
4. 显示环境为 800×600 分辨率，16 位增强色
5. delphi6.0

三、设计思路



说明：因为delphi的很多程序是已经在组件中封装好了的，只需要调用就可以了。

先是设定NMSMTP组件中要进行连接的SMTP服务器名，服务器的端口号，以及登陆用户等，再利用NMSMTP组件的Connect方法尝试连接。在进行连接过程中，可能会出现连接失败等一些错误，这里对此进行了跟踪，用来提示用户，连接是否有效。

接着开始利用NMSMTP组件指定接收方的地址，用户名，邮件主题，发送方，附件等，开始对指定的SMTP服务器进行邮件发送，由于在发送过程中可能会出现一些异常，

这里利用NMSMTh组件对在发送过程中出现的异常进行捕捉，同时反馈给用户。

为了能使邮件发送成功，在发送过程中要把邮件所包含的邮件头信息去掉，该属性的定义过程为：

```
property FinalHeader: TExStringList;
```

这里调用它的一个Clear方法用于清空邮件的头信息。

利用NMSMTh组件的OnAuthenticationFailed事件来处理需要验证的SMTP服务器，该事件在 Delphi 6.0中定义如下：

```
property onAuthenticationFailed: THandlerEvent;
```

这里的ThandlerE此止对象定义如下：

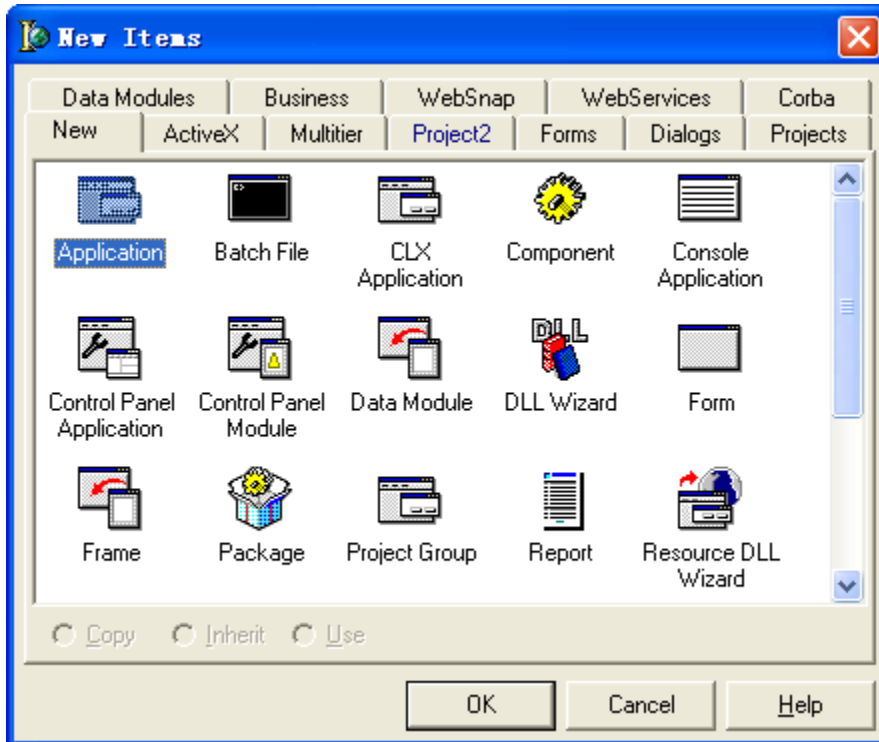
```
THandlerEvent=procedure (var Handled: Boolean) of object;
```

这里的Handled参数用于标识是否进行相应的处理。

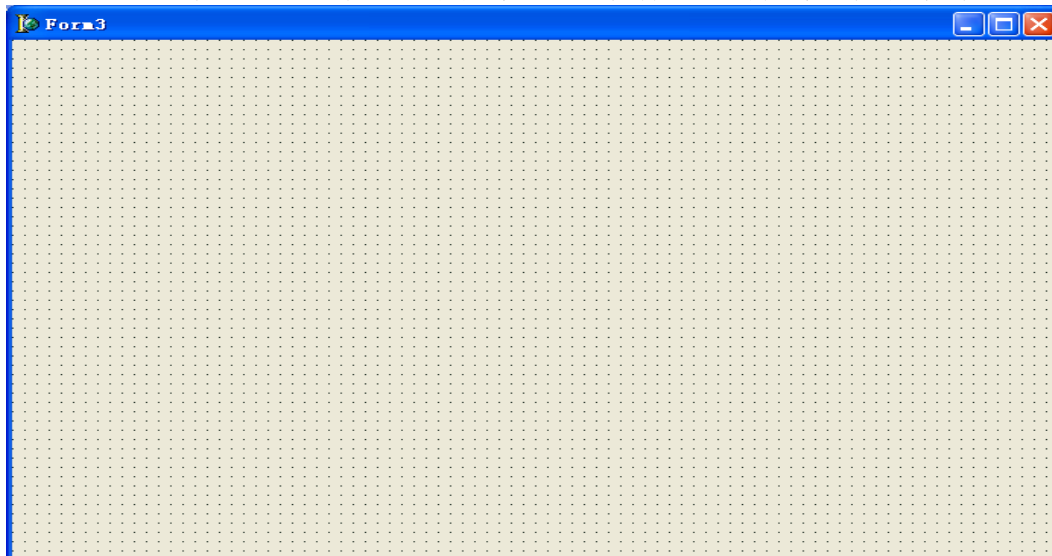
四、实验过程

1 建立一个新工程

(1)启动 Delphi 6. 0, 在 IDE环境下单击工具栏上的【New】按钮, 打开【New Items】对话框, 然后单击【New】选项卡, 这时出现了 Delphi 6. 0默认的新建选项列表, 选择Applicpon项, 如图所示:

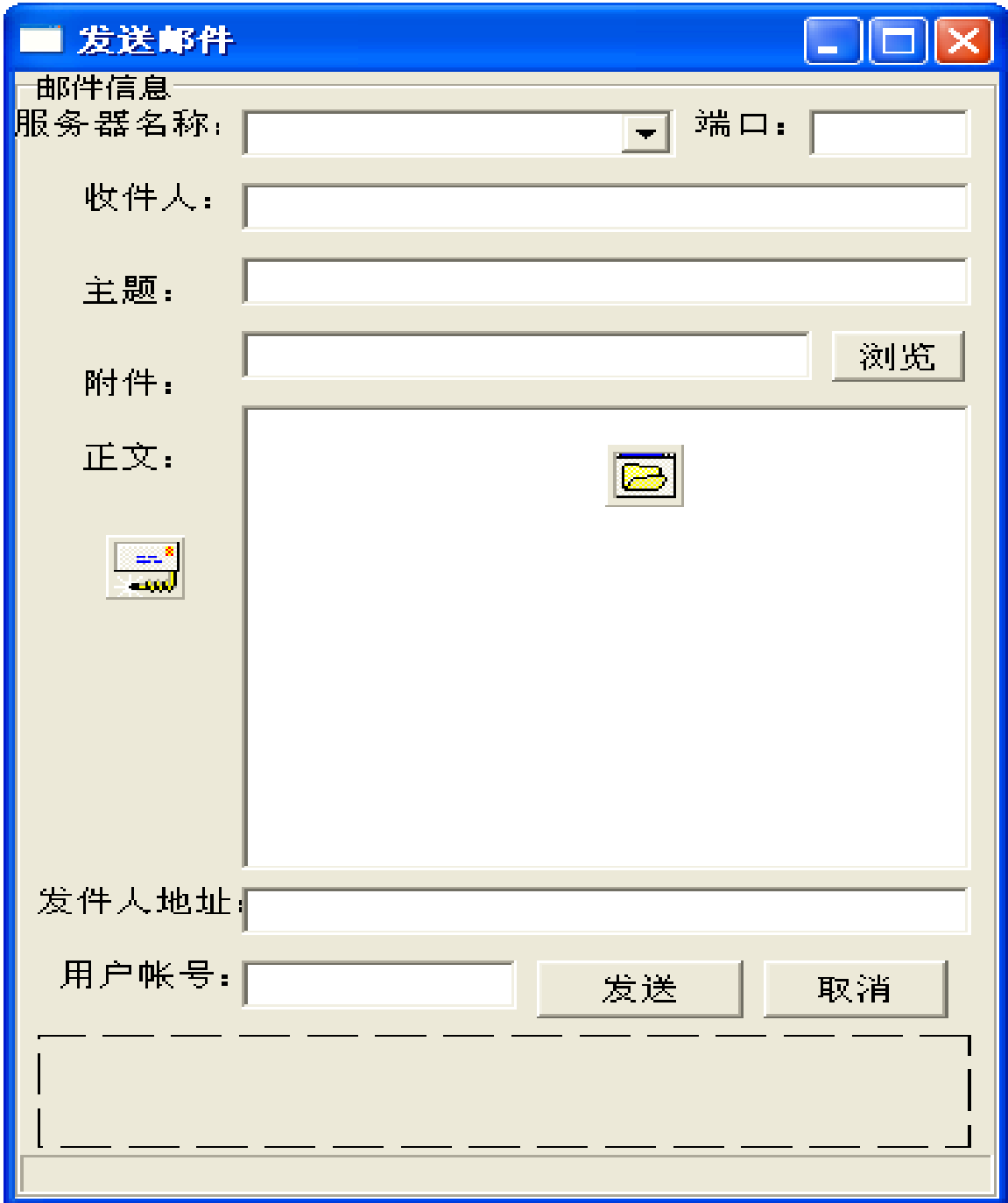


(2)单击【OK】按钮, 则创建了一个新的应用程序, 如图所示。



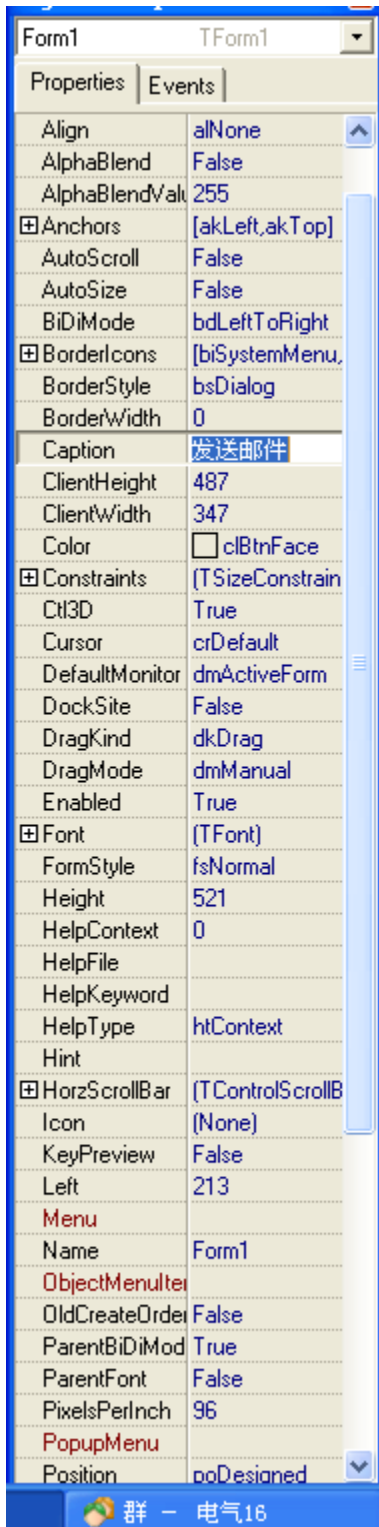
(3)现在开始构建这个发送E-mail程序的主要窗体结构, 为了能使用户可以随意指定符合自己需要的SMTP服务器, 这里提供了一个下拉文本框, 用户可以根据

自己的需要选定。用户指定好发送方和邮件主题等信息后，就可以利用该程序发送邮件了，这里所需要的组件都是由 Delphi 6. 0提供的，添加过程为：向应用程序主窗体中依次添加一个Memo组件，六个Edit组件，一个Animat组件，一个OpenDialog组件，一个NMSMTP组件，一个GroupBox组件，两个Pannel组件，一个StatusBar组件，八个Label组件和两个SpeedButton组件，具体位置如图所示



2 设置窗口属性

(1) TForm

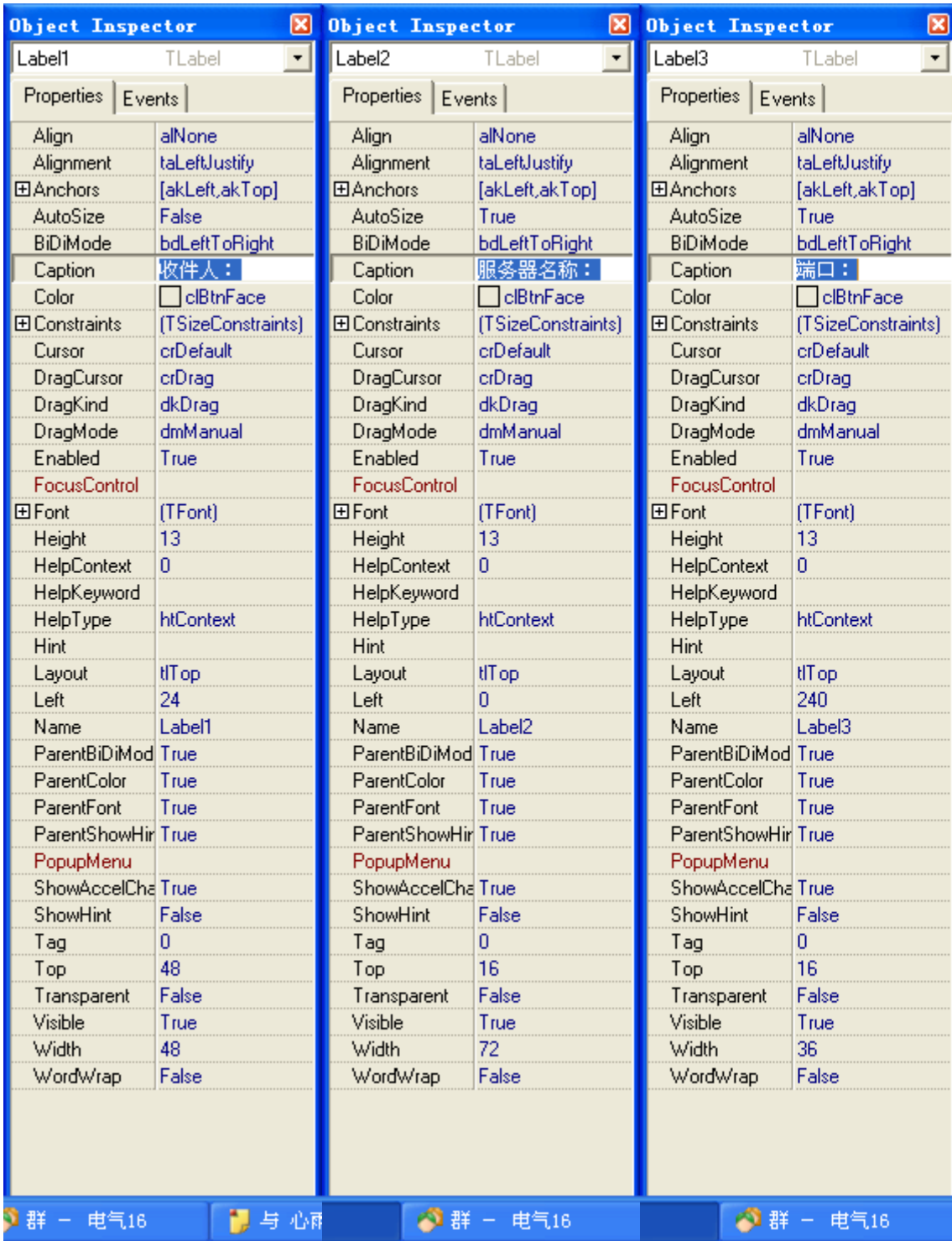


(2) TSpeedButton

The image displays three side-by-side screenshots of the Object Inspector for TSpeedButton controls. Each window shows a list of properties and their values for a specific button instance.

Property	SpeedButton1	SpeedButton2	SpeedButton3
Action			
AllowAllUp	False	False	False
⊞ Anchors	[akLeft,akTop]	[akLeft,akTop]	[akLeft,akTop]
BiDiMode	bdLeftToRight	bdLeftToRight	bdLeftToRight
Caption	浏览	发送	取消
⊞ Constraints	(TSizeConstraints)	(TSizeConstraints)	(TSizeConstraints)
Cursor	crDefault	crDefault	crDefault
Down	False	False	False
Enabled	True	True	True
Flat	False	False	False
⊞ Font	(TFont)	(TFont)	(TFont)
Glyph	(None)	(None)	(None)
GroupIndex	0	0	0
Height	22	25	25
HelpContext	0	0	0
HelpKeyword			
HelpType	htContext	htContext	htContext
Hint			
Layout	blGlyphLeft	blGlyphLeft	blGlyphLeft
Left	288	184	264
Margin	-1	-1	-1
Name	SpeedButton1	SpeedButton2	SpeedButton3
NumGlyphs	1	1	1
ParentBiDiMod	True	True	True
ParentFont	True	True	True
ParentShowHir	True	True	True
PopupMenu			
ShowHint	False	False	False
Spacing	4	4	4
Tag	0	0	0
Top	112	384	384
Transparent	True	True	True
Visible	True	True	True
Width	47	73	65

(3) Tlabel



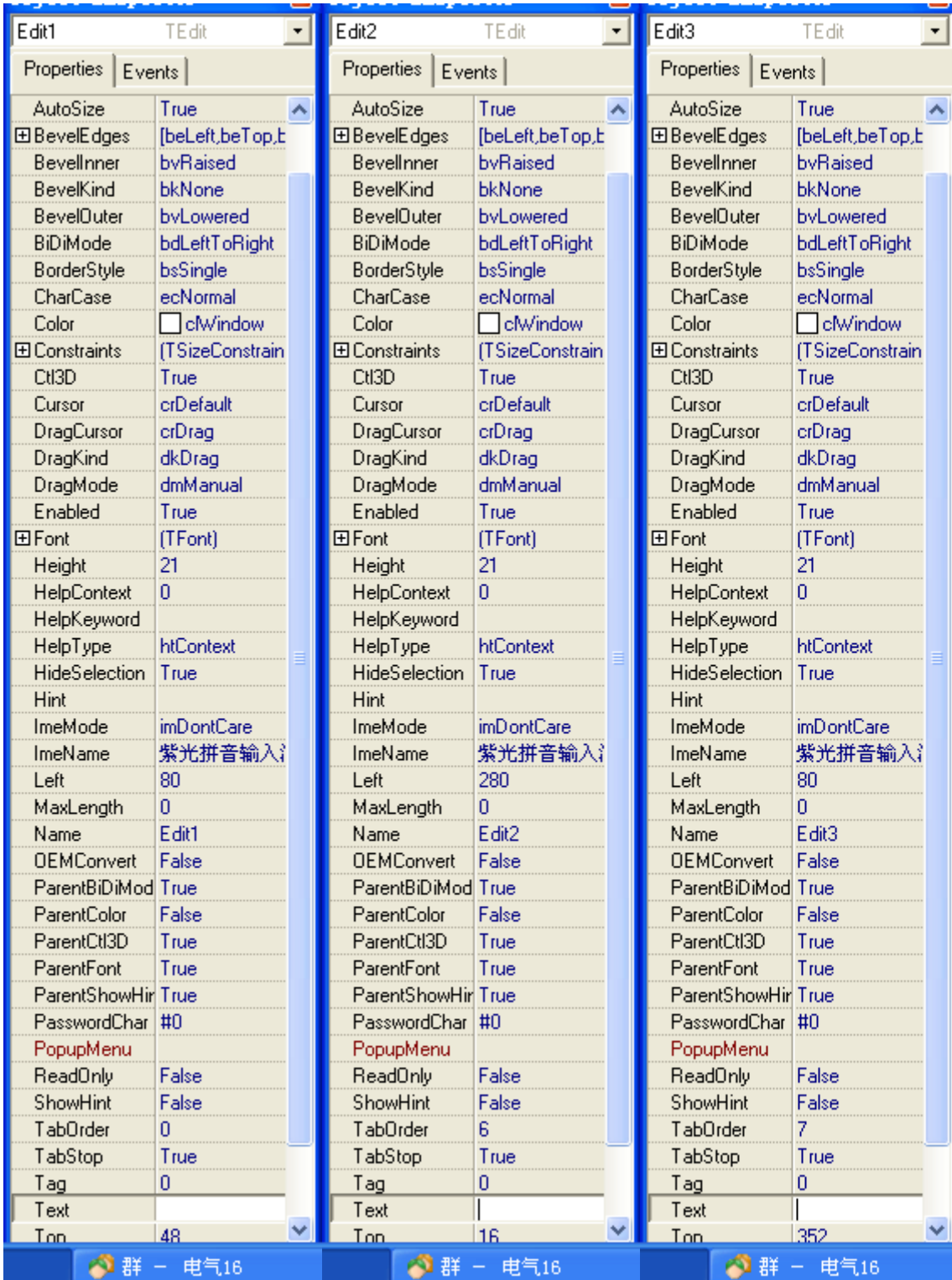
其他的label14---label18同理设置。

(3) TButton

The image displays three side-by-side screenshots of the Object Inspector window in a software development environment, each showing the properties of a TSpeedButton control. The controls are named SpeedButton1, SpeedButton2, and SpeedButton3. The properties are organized into sections: Action, Font, and PopupMenu. The 'Caption' property for each button is highlighted in blue.

Property	SpeedButton1	SpeedButton2	SpeedButton3
Action			
AllowAllUp	False	False	False
⊞ Anchors	[akLeft,akTop]	[akLeft,akTop]	[akLeft,akTop]
BiDiMode	bdLeftToRight	bdLeftToRight	bdLeftToRight
Caption	浏览	发送	取消
⊞ Constraints	(TSizeConstraints)	(TSizeConstraints)	(TSizeConstraints)
Cursor	crDefault	crDefault	crDefault
Down	False	False	False
Enabled	True	True	True
Flat	False	False	False
⊞ Font	(TFont)	(TFont)	(TFont)
Glyph	(None)	(None)	(None)
GroupIndex	0	0	0
Height	22	25	25
HelpContext	0	0	0
HelpKeyword			
HelpType	htContext	htContext	htContext
Hint			
Layout	blGlyphLeft	blGlyphLeft	blGlyphLeft
Left	288	184	264
Margin	-1	-1	-1
Name	SpeedButton1	SpeedButton2	SpeedButton3
NumGlyphs	1	1	1
ParentBiDiMod	True	True	True
ParentFont	True	True	True
ParentShowHir	True	True	True
PopupMenu			
ShowHint	False	False	False
Spacing	4	4	4
Tag	0	0	0
Top	112	384	384
Transparent	True	True	True
Visible	True	True	True
Width	47	73	65

(4) TEdit



其他同理。

(5) TGroupBox

GroupBox1 TGroupBox	
Properties	Events
Align	alClient
⊞ Anchors	[akLeft,akTop,akRight]
BiDiMode	bdLeftToRight
Caption	邮件信息
Color	<input type="checkbox"/> clBtnFace
⊞ Constraints	(TSizeConstraints)
Ctl3D	True
Cursor	crDefault
DockSite	False
DragCursor	crDrag
DragKind	dkDrag
DragMode	dmManual
Enabled	True
⊞ Font	(TFont)
Height	487
HelpContext	0
HelpKeyword	
HelpType	htContext
Hint	
Left	0
Name	GroupBox1
ParentBiDiMod	True
ParentColor	True
ParentCtl3D	True
ParentFont	True
ParentShowHir	True
PopupMenu	
ShowHint	False
TabOrder	0
TabStop	False
Tag	0
Top	0
Visible	True
Width	347

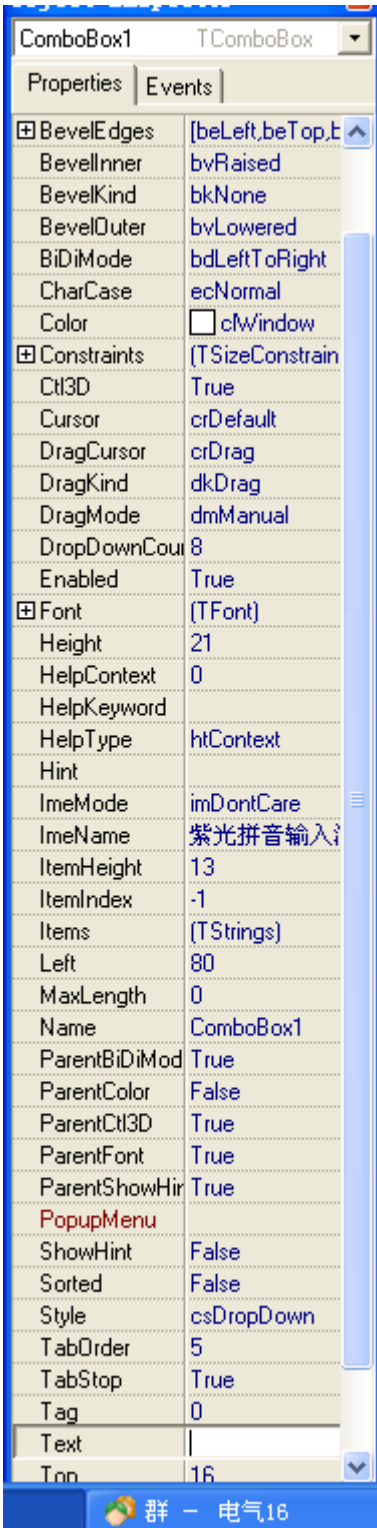
(6) Tmemo

Memo1 Tmemo	
Properties	Events
Align	alNone
Alignment	taLeftJustify
⊞ Anchors	[akLeft,akTop]
⊞ BevelEdges	[beLeft,beTop,beBottom,beRight]
BevelInner	bvRaised
BevelKind	bkNone
BevelOuter	bvLowered
BiDiMode	bdLeftToRight
BorderStyle	bsSingle
Color	<input type="checkbox"/> clWindow
⊞ Constraints	(TSizeConstraints)
Ctl3D	True
Cursor	crDefault
DragCursor	crDrag
DragKind	dkDrag
DragMode	dmManual
Enabled	True
⊞ Font	(TFont)
Height	201
HelpContext	0
HelpKeyword	
HelpType	htContext
HideSelection	True
Hint	
ImeMode	imDontCare
ImeName	紫光拼音输入法
Left	80
Lines	(TStrings)
MaxLength	0
Name	Memo1
OEMConvert	False
ParentBiDiMod	True
ParentColor	False
ParentCtl3D	True
ParentFont	True
ParentShowHir	True
PopupMenu	
ReadOnly	False
ScrollBars	ssNone
ShowHint	False
TabOrder	2
TabStop	True
Tag	0

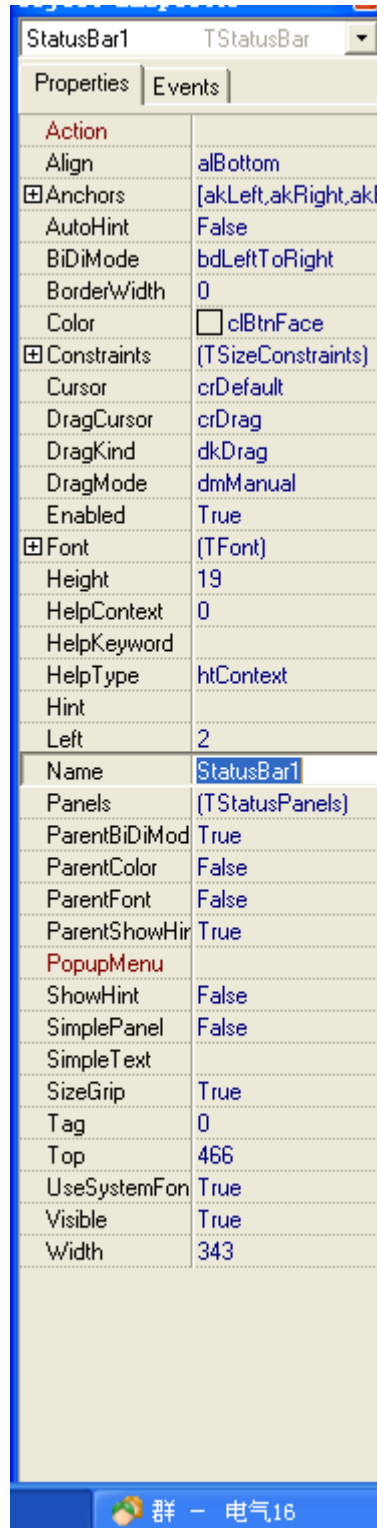
(7) TAnimate

Animate1 TAnimate	
Properties	Events
Active	False
Align	alNone
⊞ Anchors	[akLeft,akTop]
AutoSize	False
BorderWidth	0
Center	True
Color	<input type="checkbox"/> clBtnFace
CommonAVI	aviCopyFile
⊞ Constraints	(TSizeConstraints)
Cursor	crDefault
FileName	
Height	49
HelpContext	0
HelpKeyword	
HelpType	htContext
Hint	
Left	8
Name	Animate1
ParentColor	True
ParentShowHir	True
Repetitions	0
ShowHint	False
StartFrame	1
StopFrame	26
Tag	0
Timers	False
Top	416
Transparent	True
Visible	True
Width	329

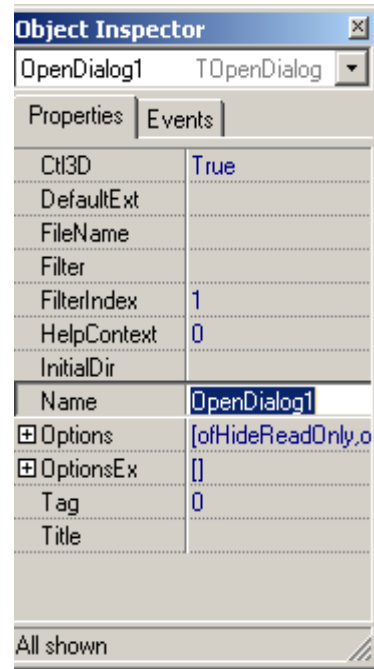
(8) TcomboBox



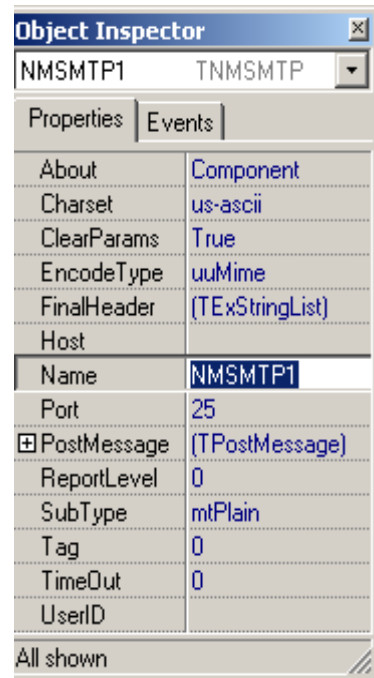
(9) TStatusBar



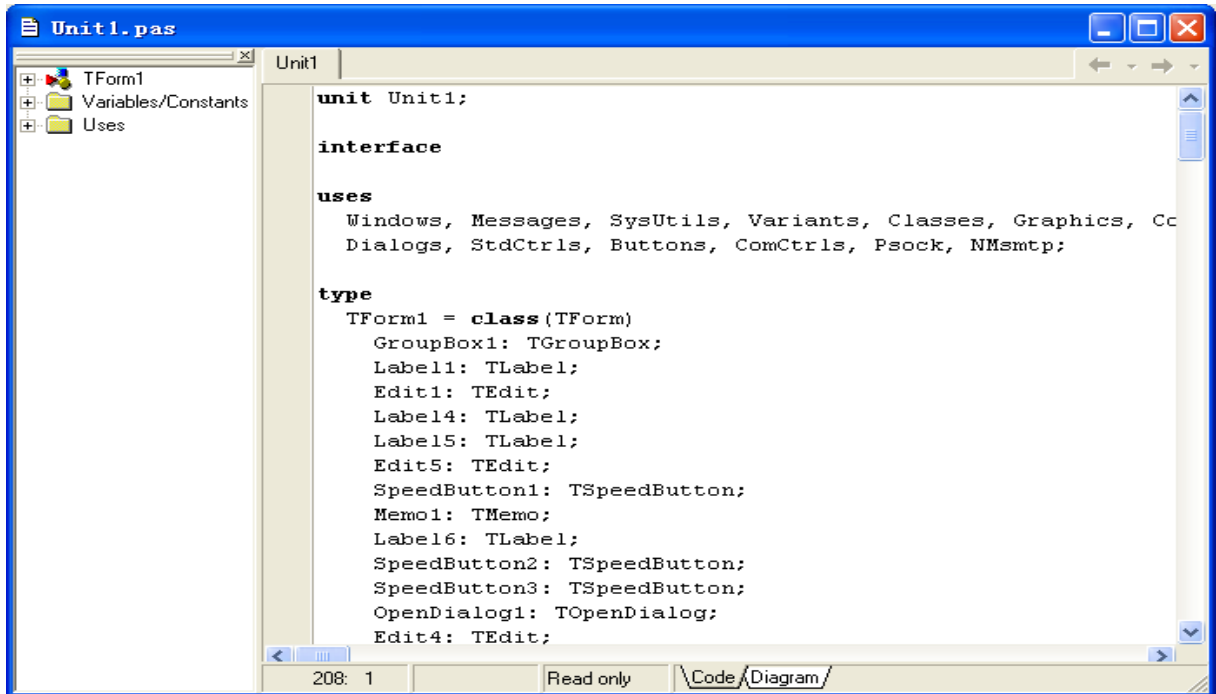
(10) TOpenDialog



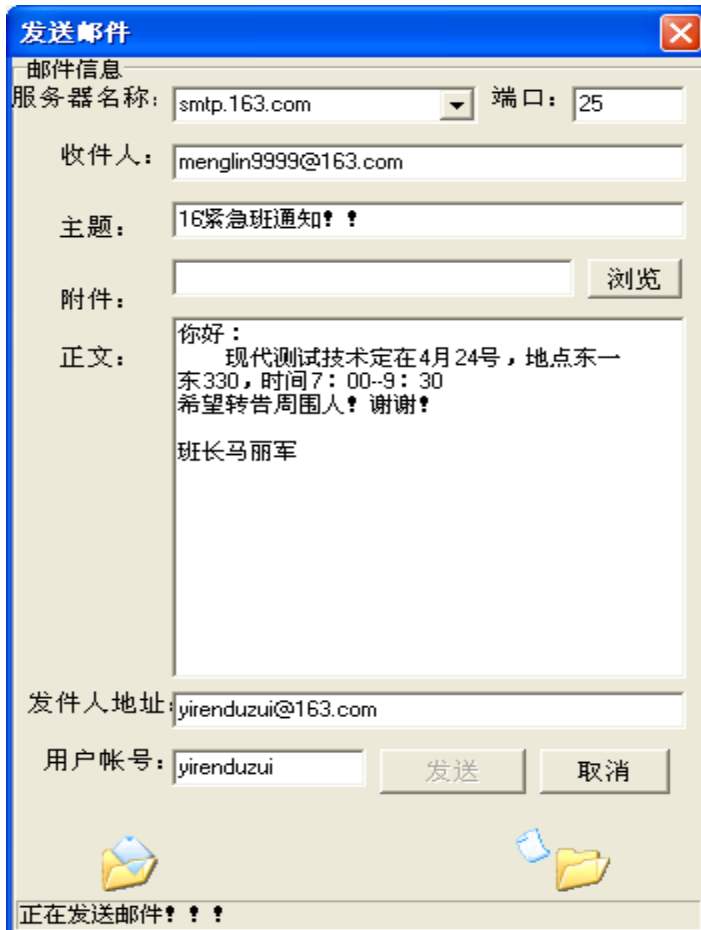
(11) TNMSMTP



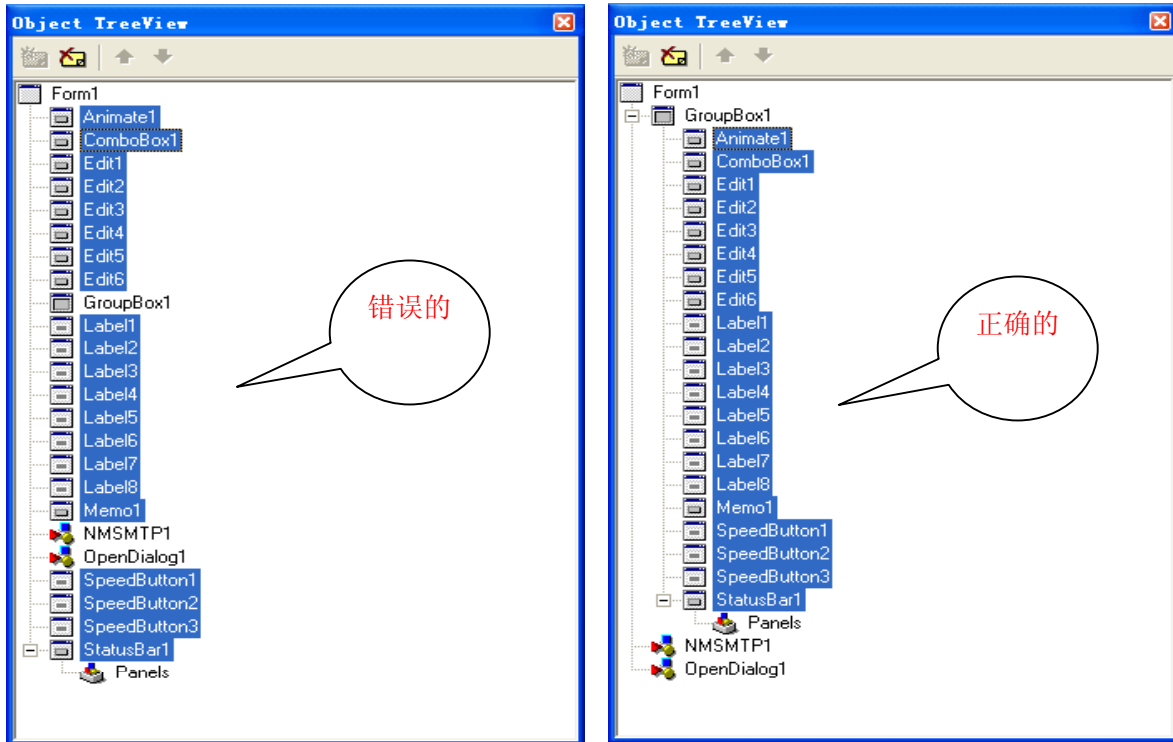
3添加应用程序源代码:



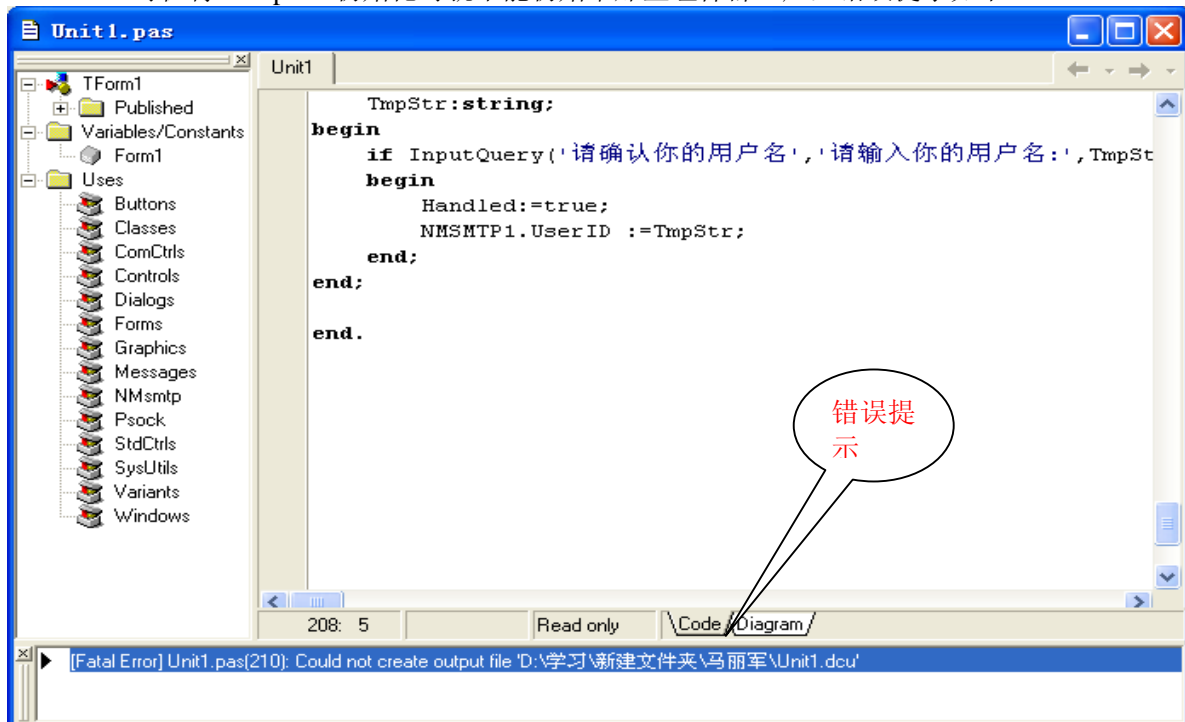
运行结果



五、实验故障与排除



较可以知道，我把所有的属于GroupBox的组件群放在了他的外面成为同等的组件，这样在运行时在将GroupBox初始化时就不能初始华那些组件群。产生错误提示如下：



我后来在对一个组件跟踪调试的时候才发现未能初始化，依次类推都没有初始化，后改变他们的位置就编译通过了。

六、个人总结

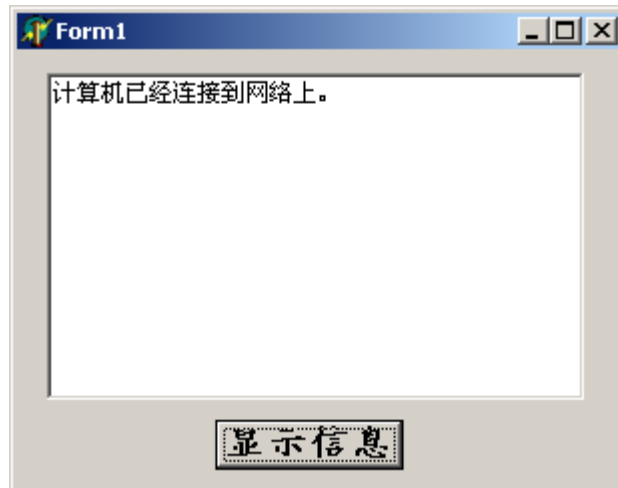
历时 4 周，我终于完成了这个实验，因为自己的计算机水平相对较低，所以有一半的时间是用来熟悉 delphi 和多网络的控制。熟悉了这些以后，自己做起来就相对容易些了。

通过此次的程序制作,我对 delphi 的应用和计算机邮件的发送过程有了一定深入的了解,从以前的一知半解到现在可以运用它做出自己满意的东西。在制作的过程中曾经有过整日调程序,整天泡图书馆的痛苦,也有过跑遍宿舍楼问周围会使用 delphi 的同学,更大的的是有丰收的喜悦。其实任何事物只要自己肯努力去研究,就总是会有收获的。

以前编程在我看来是很难的一件事,但是经过这次的学习,我发现,虽然它很难,但只要你用心去学,就一定会有收获。以前觉得网络很神秘,它能够运行起来真是奇迹,而现在却了解他里面的运做也实现,更多感觉网络世界和人类世界是多么的相同(呵呵,好象是废话,他们就是人类创造出来的),他们也使用协议,也有规则,也象一个个生命体的存在一样。我本人也对网络产生了更大的兴趣,对于我们电气的学生将来好多要面对的是电网的连接,计算机控制网络连接,也有可能是计算机网络和电网的连接,这些都是我们将来要考虑的问题,所以现在应该学好网络的知识,为将来打下扎实的基础。

我在这次实验后虽然还有好多的不明白的地方,但感觉收获很大,不明白的自己以后可以继续学习。自己又尝试制作了一个测试计算机是否连接的程序(附图)。其实学什么东西都可以举一反三的道理是相通的。

感谢老师能给提供一个这样的机会和在实验过程中对自己的指导,也感谢周围人对自己的帮助!



附录

程序代码:

(1) 初始化

这里主要完成窗体中的一个属性的设置工作,同时完成对主窗口的操做,这里主要完成窗口类型定义,窗口初始化,初始位置等属性设定,还声明了将要用的变量 要调用的事件。

代码:

```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, Buttons, ComCtrls, Psock, NMsmtplib;  
type  
  TForm1 = class(TForm)
```



```

GroupBox1: TGroupBox;
Label1: TLabel;
Edit1: TEdit;
Label4: TLabel;
Label5: TLabel;
Edit5: TEdit;
SpeedButton1: TSpeedButton;
Memo1: TMemo;
Label6: TLabel;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
OpenDialog1: TOpenDialog;
Edit4: TEdit;
NMSMTP1: TNMSMTP;
Label2: TLabel;
Animate1: TAnimate;
ComboBox1: TComboBox;
Label3: TLabel;
Edit2: TEdit;
Label7: TLabel;
Edit3: TEdit;
Label8: TLabel;
Edit6: TEdit;
StatusBar1: TStatusBar;
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure NMSMTP1ConnectionFailed(Sender: TObject);
procedure NMSMTP1SendStart(Sender: TObject);
procedure NMSMTP1Success(Sender: TObject);
procedure NMSMTP1Connect(Sender: TObject);
procedure NMSMTP1Disconnect(Sender: TObject);
procedure NMSMTP1InvalidHost(var Handled: Boolean);
procedure NMSMTP1Failure(Sender: TObject);
procedure NMSMTP1AuthenticationFailed(var Handled: Boolean);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

```

(2) 发通电子邮件

发送邮件使用的是 Delphi 6 的组件 TNMSMTP，在此组件中封装了发送邮件的种种

属性、方法和事件，这部分的代码实现的就是利用该组件来发送电子邮件。

程序先要判断 NMSMTP 组件是否已经连上服务器，避免多次连接，浪费资源。接下来的工作是设置服务器的端口然后进行一些发送前的邮件发送条件的判断工作，检测是否有合法的电子邮件地址，是否有标题等，判断附件是否为空，进行发送，并提示邮

件发送结果，最后断开与服务器的连接。

具体的实现代码如下：

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    Edit5.Text := ''; //清空文本框
    if Opendialog1.Execute then //打开 WINDOWS 窗口
        Edit5.Text := Opendialog1.FileName; //获取文件名
        //判断如果为空，则退出当前程序
        if Edit5.Text = '' then Exit;
end;
procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    //判断是否确定要退出该应用程序
    if MessageDlg('确定要退出吗?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
        Close; //退出应用程序
end;
procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    Animate1.Active := false; //动画停止
    if NMSMTP1.Connected then //判断 NMSMTP 组件是否已经连上服务器
        NMSMTP1.Disconnect; //避免多次连接，浪费资源
    if (Edit1.Text = '') then //判断收件人的邮件地址是否为空
        ShowMessage('请输入收件人邮件地址') //提示信息
    else
        if (Edit4.Text = '') then //判断邮件主题是否为空
            ShowMessage('请输入邮件主题') //提示信息
        else
            //判断发送邮件的内容是否为空
            if (Memo1.Lines.Count > 0) then
                begin
                    //动画开始运行
                    Animate1.Active := true;
                    //按钮为非可用状态
                    SpeedButton2.Enabled := false;
                    //发送邮件的主机名
                    NMSMTP1.Host := Combobox1.Text;
                    //发送邮件的端口
                    NMSMTP1.Port := strtoint(Edit2.Text);
                    //用户 ID 号
                    NMSMTP1.UserID := Edit6.Text;
                    try
                        //连接发送邮件服务器
                        NMSMTP1.Connect;
                        StatusBar1.Update;
                    except
                        begin
                            //提示服务器连接失败
                            ShowMessage('连接服务器失败!!! ');
                            //动画停止播放
                        end;
                    end;
                end;
            end;
        end;
end;
```

```

        Animate1.Active:=False;
        //按钮处于激活状态
        SpeedButton2.Enabled :=true;
        //取消发送该邮件
        Exit;
    end;
end;
//设置发件人地址
NMSMTP1.PostMessage.FromAddress :=Edit3.Text;
//设置收件人地址
NMSMTP1.PostMessage.ToAddress.Add(Edit1.Text);
//设置发送邮件的主题
NMSMTP1.PostMessage.Subject :=Edit4.Text;
//设置发送邮件的时间
NMSMTP1.PostMessage.Date :=Datetostr(Now);
//判断附件是否为空
if Edit5.Text <>" then
    NMSMTP1.AppendFile(Edit5.Text); //携带附件
//携带邮件正文
NMSMTP1.PostMessage.Body.Assign(Memo1.Lines);
//传输邮件类型
NMSMTP1.EncodeType :=uuMime;
NMSMTP1.PostMessage.FromName :=Edit6.Text;
try
    begin
        StatusBar1.SimpleText :='正在发送邮件!!! ';
        NMSMTP1.SendMail; //发送邮件
    end;
except
    begin
        //提示邮件发送失败
        ShowMessage('发送邮件失败!!! ');
        //断开发送邮件服务器的连接
        NMSMTP1.Disconnect;
        //动画停止
        Animate1.Active:=False;
        //按钮处于激活状态
        SpeedButton2.Enabled :=true;
        //取消发送
        Exit;
    end;
end;
Animate1.Active:=False; //动画停止
SpeedButton2.Enabled :=true;//按钮处于激活状态
Edit1.Text :="; //清空文本框
Edit4.Text :=";
Edit5.Text :=";
Memo1.Clear;
end;
end;

```

(3) 处理各组件的各种状态:

```
procedure TForm1.NMSMTP1ConnectionFailed(Sender: TObject);
begin
    //邮件服务器发送失败时, 提示邮件发送失败
    ShowMessage('连接服务器失败!!! ');
end;

procedure TForm1.NMSMTP1SendStart(Sender: TObject);
begin
    //用于删除邮件中所有的头部信息
    NMSMTP1.FinalHeader.Clear;
end;

procedure TForm1.NMSMTP1Success(Sender: TObject);
begin
    //邮件发送成功后提示邮件发送成功
    StatusBar1.SimpleText :='邮件发送成功!!! ';
end;

procedure TForm1.NMSMTP1Connect(Sender: TObject);
begin
    StatusBar1.SimpleText :='正在连接邮件服务器!!! ';
end;

procedure TForm1.NMSMTP1Disconnect(Sender: TObject);
begin
    StatusBar1.SimpleText :='断开与邮件服务器的连接!!! ';
end;

procedure TForm1.NMSMTP1InvalidHost(var Handled: Boolean);
begin
    StatusBar1.SimpleText :='邮件服务器名错误, 发送失败!!! ';
end;

procedure TForm1.NMSMTP1Failure(Sender: TObject);
begin
    StatusBar1.SimpleText :='邮件发送失败!!! ';
end;

procedure TForm1.NMSMTP1AuthenticationFailed(var Handled: Boolean);
var
    TmpStr:string;
begin
    if InputQuery('请确认你的用户名','请输入你的用户名:',TmpStr) then;
    begin
        Handled:=true;
        NMSMTP1.UserID :=TmpStr;
    end;
end;
end.
```

TNMSMTP 控件的常用属性

Host 属性

指定 SMTP 邮件服务器的地址, 如 SMTP. SIM. COM. CN, 该属性定义如下:

```
property Host: String;
```

Port 属性

指定 SMTP 邮件服务器的端口号, 一般用其默认值 25 即可, 该属性定义如下:

property Port: Integer;

. Connected 属性

判断客户端程序是否和 SMTP 邮件服务器正确连接, 该属性定义如下:

property Connected: boolean;

. BeenCanceled 属性

判断当前操作是否被取消, 当属性值为 False 时表示当前操作没有被取消, 该属性定义如下:

property BeenCanceled: boolean;

. BeenTimedout 属性

判断当前操作是否发生延时错误, 该属性定义如下:

property BeenTimedout: boolean;

. BytesRecv 属性

表示收到的字节数, 该属性定义如下:

property BytesRecv: IOFlags;

. BytesSent 属性

表示收到发送的字节数, 该属性定义如下:

property BytesSent: longInt;

. BytesTotal 属性

表示发送的和收到的字节总数, 该属性定义如下:

property BytesTotal: longInt;

. UserID 属性

指定用户名, 当与邮件服务器建立连接时, 需验证用户名, 该属性定义如下:

property UserID: string;

. PostMessage 属性

包含将要被发送的邮件内容, 它有几个附属的子属性, 该属性定义如下:

property PostMessage: TPostMessage;

. PostMessage.FromAddress 属性

发件人地址, 该属性定义如下:

property PostMessage.FromAddress: string;

PostMessage.ToAddress 属性

收件人地址, 该属性定义如下:

property PostMessage.ToAddress: TStringList;

. PostMessage.Attachments 属性

邮件附件的文件列表, 该属性定义如下:

property PostMessage.Attachments: TStringList;

PostMessage.Body 属性

邮件的正文, 该属性定义如下:

property PostMessage.Body: TStringList;

. PostMessage.Subject 属性

邮件的主题, 该属性定义如下:

property PostMessage.Subject: TStringList;

. PostMessage.Date 属性

邮件的发送日期, 该属性定义如下:

property PostMessage.Date: string;

TN MSMTTP 控件的常用方法

. Connect 方法

建立与远程 SMTP 服务器的连接, 连接前需指定 Host 和 UserID, 该方法的函数原型

如下所示:

property Connect; virtual;

Disconnect 方法

断开与远程服务器之间的连接, 该方法的函数原型如下所示:

procedure Disconnect; virtual;

SendMail 方法

将当前邮件发送到远程服务器, 该方法的函数原型如下所示:

procedure SendMail;

Listen 方法

从 Port 属性指定的端口监听 TCP 连接, 该方法的函数原型如下所示:

procedure Listen (sync: boolean);

Cancel 方法

取消当前的输入输出操作并断开到远程服务器的连接, 该方法的函数原型如下所示:

procedure Cancel;

SendFile 方法

将当前邮件所包含的文件发送到远程服务器, 该方法的函数原型如下所示:

procedure sendFile (FileName: String);

. TmsactiOn 方法

将邮件命令发送到远程服务器, 并返回服务器的反馈信息, 该方法的函数原型如下所示:

function Transaction (constCommandstring: string): String; virtual;

. Accept 方法

接受远程计算机的连接请求, 该方法的函数原型如下所示:

function Accept: Word; virtual;

@Write 方法

从当前的连接只读取一定数目的数据, 并把它们转换成字符串, 该方法的函数原型如下所示:

function read (value: word): string;