

学用软件起步与进阶丛书

Delphi 5 网络编程

起步与进阶

黄亮 黄军万 编著

国防工业出版社



100

TP311.56

H76c

学用软件起步与进阶丛书

Delphi 5 网络编程 起步与进阶

黄 亮 黄军万 编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

Delphi 5 网络编程起步与进阶/黄亮,黄军万编著.
北京:国防工业出版社,2000.10
(学用软件起步与进阶丛书)
ISBN 7-118-02252-7

I . D… II . ①黄… ②黄… III . DELPHI 语言-程序设计 IV . TP312

中国版本图书馆 CIP 数据核字(2000)第 33999 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 20½ 471 千字

2000 年 10 月第 1 版 2000 年 10 月北京第 1 次印刷

印数:1-3000 册 定价:28.00 元

(本书如有印装错误,我社负责调换)

前 言

Borland 公司已经向全球发布了其新一代的可视化编程软件——Delphi 5。Delphi 是 Borland 公司于 1995 年推出的 Windows 环境下可视化编程语言,在短短两年中已经发行了 60 万份以上,目前国内流行的版本为 3.0X 和 4.0X(32 位)。最近 Borland 又推出了功能更为强大的 Delphi 5,以及与 Delphi 一样采用 VCL 构架的 C++ Builder。经著名的 PC Week Labs 测试证明:Delphi 5 包括了 Visual C++ 5.0 与 Visual Basic 5.0 的功能和优点,而且不像微软公司刚推出的 Visual Studio 工具组合只提供个别、单一的开发环境,Delphi 5 提供较佳的环境与执行速度,其速度比刚推出的 Visual Basic 5.0 快了 3~6 倍以上。的确,Delphi 将我们从复杂的 Windows 编程中解放出来,极大地提高了编程效率。Delphi 5 适用于 Windows NT 和 Windows 95/98/2000 环境,新版本对旧有版本的 Delphi 在诸多性能和功能方面都有所改善。

Delphi 被称为是第四代编程语言,它是基于窗口和面向对象的编程方法,与 Windows 操作系统紧密结合,具有强大的数据库技术支持,迅捷的编译速度。同时还兼备了 C 语言的强大功能和 Visual Basic 易学灵活的特点,一直是程序员们喜爱的编程工具。Delphi 特别适合于编制中型应用程序,一些通用的网络程序都是用它开发出来的。鉴于 Delphi 的特性,为顺应广大用户的要求,作者在积累大量网络开发经验的基础上编写了这本书。

本书包括 8 章:第 1 章介绍了收发 E-mail 的程序;第 2 章介绍了网络聊天程序;第 3 章介绍了 FTP 下载软件的编制;第 4 章介绍了网页浏览器程序的编制;第 5 章介绍了 COM 概念及其接口实现;第 6 章介绍了如何创建 Automation 服务器和控制器;第 7 章介绍了如何创建 ActiveX 控件及其在网络上的发布;第 8 章介绍了如何利用 Delphi 5 提供的 WebBroker 技术创建交互式的动态网页。

通过本书的学习,读者将对控件编程和面向对象编程有一个较深入的认识。在本书指导下,读者在完成程序的设计之后,将具备中级 Delphi 编程水平。

本书主要由黄亮和黄军万编著,同时参与编写的人员还有薛小香、李春和、冯树红、陈立、邓冰、梁贵、李峰、李灯、贺梧州以及黎明等。

在本书的编写过程中,得到了林章波同志的大力支持,对本书提出了许多中肯的意见,在此表示衷心的感谢。

编 著 者

第 1 章

邮 政 信 箱

电子邮件是当今人们相互沟通信息的主要方式,它以其便捷的方式迅速取代了传统的纸张信笺。本章主要介绍 NMSMTP 控件、NMPOP3 控件的使用方法,使读者在一天之内对电子邮件工具的设计过程有一个初步的认识。

本章具体包括以下内容:

- 控件位置
- 界面设计
- 程序编制
- NMSMTP 控件主要属性、方法、事件
- NMPOP3 控件主要属性、方法、事件

1.1 设计邮政信箱

1.1.1 控件位置

电子邮件是当今人们相互沟通信息的主要方式,它以其便捷的方式迅速取代了传统的纸张信笺。目前,在个人计算机上比较流行的电子邮件工具是 Exchange、Internet Explorer 电子邮件、Eudora、Foxmail 等软件。在 Internet 上,电子邮件工具是必不可少的,编写具有个人风格的电子邮件更能增加用户的兴趣。下面,将给读者一步一步地讲解邮政信箱的设计过程,并给予详细的提示和说明,使读者在一天之内对电子邮件的设计过程有一个初步的认识。

首先让我们熟悉一下邮政信箱所需要用到控件的位置。在设计中将使用如下几个控件:NMSMTP 控件、NMPPOP3 控件、OpenDialog 控件、Button 控件、Memo 控件、Edit 控件、Label 控件、PageControl 控件、StatusBar 控件。

其中 NMSMTP 控件、NMPPOP3 控件在【控件】工具栏的 FastNet 选项卡中,如图 1.1 所示。

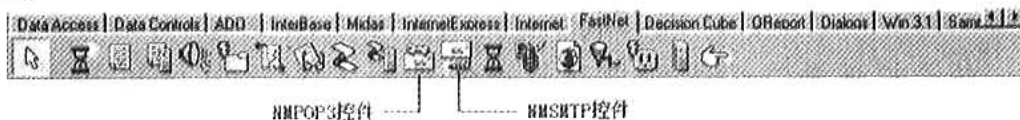


图 1.1 NMSMTP 控件、NMPPOP3 控件的位置

其中 Button 控件、Memo 控件、Edit 控件、Label 控件在【控件】工具栏的 Standard 选项卡中,如图 1.2 所示。

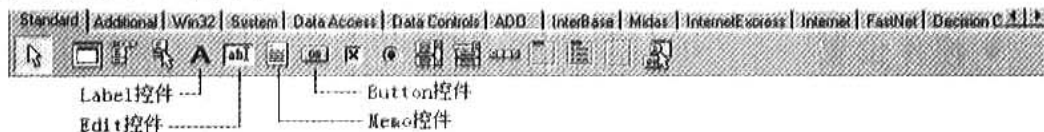


图 1.2 Button 控件、Memo 控件、Edit 控件、Label 控件的位置

其中 StatusBar 控件、PageControl 控件在【控件】工具栏的 Win32 选项卡中,如图 1.3 所示。

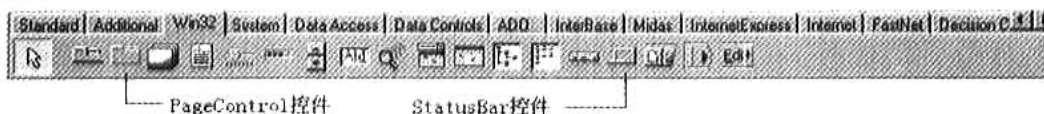


图 1.3 StatusBar 控件、PageControl 控件的位置

其中 OpenDialog 控件在【控件】工具栏的 Dialogs 选项卡中,如图 1.4 所示。

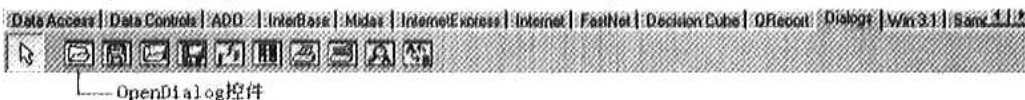


图 1.4 OpenDialog 控件的位置

1.1.2 界面设计

在掌握了各个控件的位置以后,现在就可以开始着手进行邮政信箱的界面设计工作了。打开 File 菜单,单击 New 选项,弹出 New Items 对话框,如图 1.5 所示。



图 1.5 New Items 对话框

用鼠标单击 Application 图标,并单击 OK 按钮,建立新的工程,如图 1.6 所示。

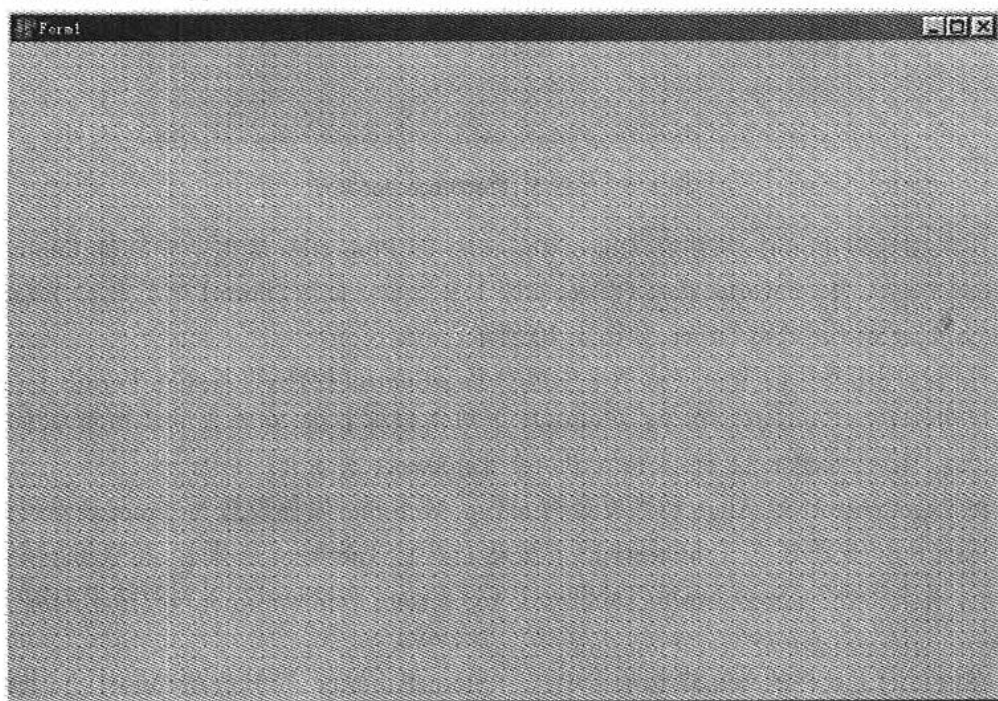


图 1.6 开始建立新的工程

打开 File 菜单,选择 Save 选项,保存新的工程文件,分别存为 main.pas 和 MyPostOffice.dpr,如图 1.7 和 1.8 所示。



图 1.7 保存 main.pas 文件



图 1.8 保存 MyPostOffice.dpr 文件

用鼠标选中 Form1,然后在 Object Inspector 的 Properties 选项卡中选择 Caption 选项,在其中键入 My Favorite Post Office,如图 1.9 所示。此时,Form1 的标题栏中的标题变为 My Favorite Post Office,如图 1.10 所示。

在【控件】工具栏的 Win32 选项卡中选中 PageControl 控件,将其放在 Form1 上,其缺省名字叫做 PageControl1。在 PageControl1 上单击鼠标右键,在弹出的快捷菜单中选择 New Page 选项,如图 1.11 所示,建立了一个 TabSheet1 选项卡。

将 PageControl1 的 Align 属性设置为 alTop, Multiline 属性设置为 True(表示可以有多排选项卡)。按照建立 TabSheet1 的方法建立其它 TabSheet 选项卡,其 Name 属性和 Caption 属性分别为:Send Email/TabSheet1、Get Email/TabSheet2,得到的结果如图 1.12 所示。

在【控件】工具栏的 Win32 选项卡中选中 StatusBar 控件,将其放在 Form1 上,其缺省名字叫做 StatusBar1。

下面我们逐一介绍各个选项卡的设计。选中 TabSheet1(属于 Send Email 选项卡),在里面添加如下一个 GroupBox,将它的 Caption 属性和 Name 属性都设置为 Connection。在 Connection 中添加如下几个控件:3 个 Label 控件,3 个 Edit 控件,2 个 Button 控



图 1.9 键入“My Favorite Post Office”

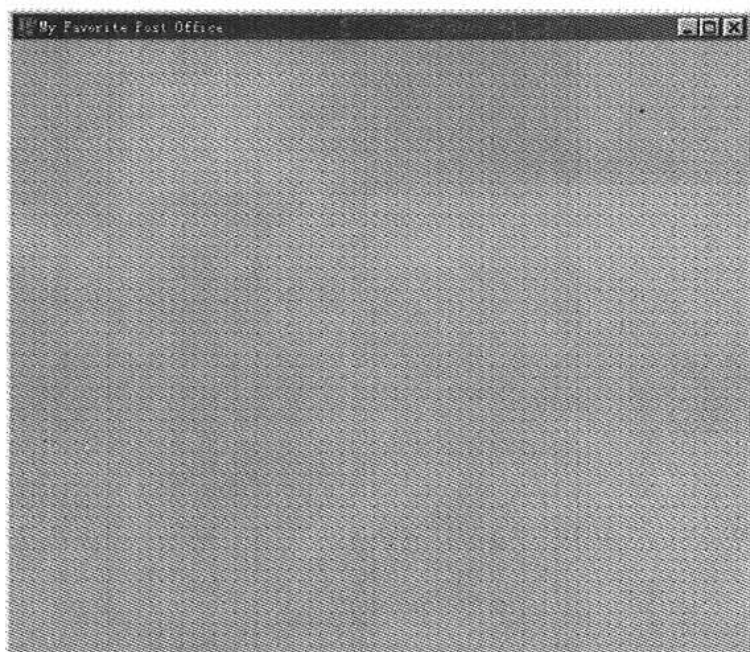


图 1.10 标题变成 My Favorite Post Office

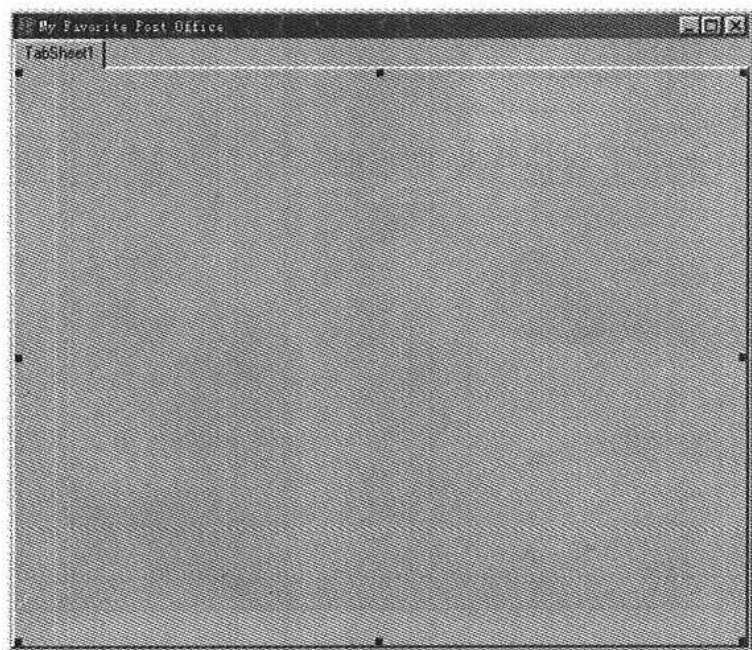


图 1.11 建立了一个 TabSheet1 选项卡

件。其中 3 个 Label 控件的 Caption 属性分别为 Remost Host、Port、User Name; 3 个 Edit 控件的 Text 属性都为空; 2 个 Button 控件的 Caption 属性和 Name 属性分别为 Connect/Connect1、Disconnect/Disconnect1, 设计结果如图 1.13 所示。

接着在 TabSheet1(属于 Send Email 选项卡)添加第二个 GroupBox, 将它的 Caption 属性和 Name 属性设置为 Personal Information/Personal Information。在 Personal Informa-

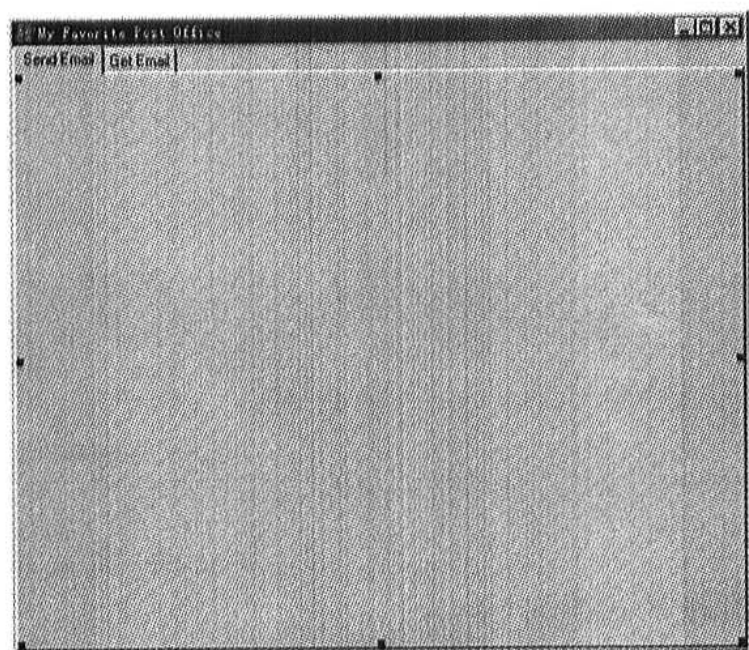


图 1.12 设计选项卡

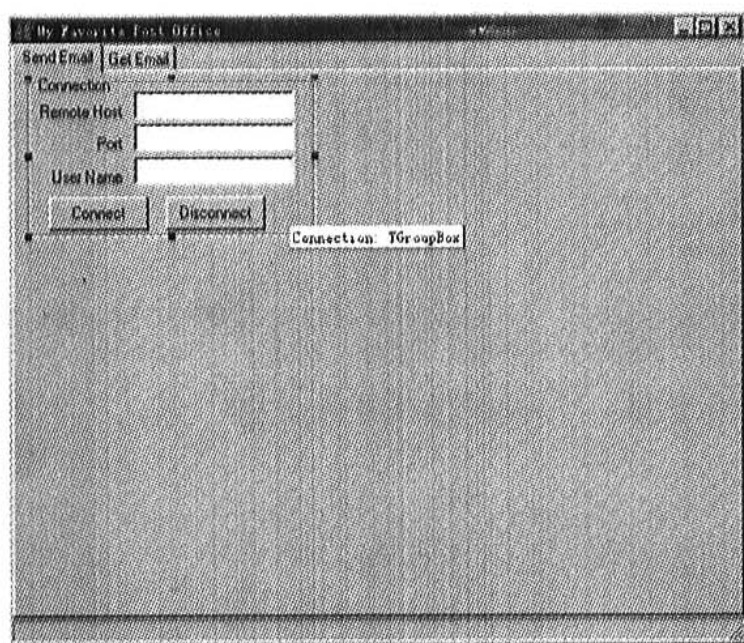


图 1.13 设计 Connection

tion 中添加如下几个控件:2 个 Label 控件,2 个 Edit 控件。其中 2 个 Label 控件的 Caption 属性分别为 Name、Email,用来给用户输入自己的地址信息;2 个 Edit 控件的 Text 属性都为空,设计结果如图 1.14 所示。

在 TabSheet1(属于 Send Email 选项卡)添加第三个 GroupBox,将它的 Caption 属性和 Name 属性都设置为 Email to Send/GroupBox1。在 Email to Send 中添加如下几个控件:5

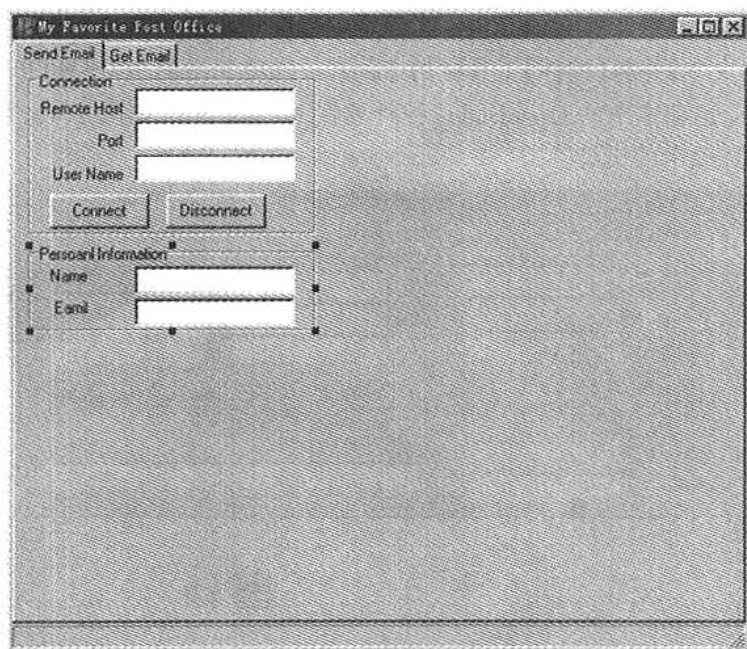


图 1.14 设计 Personal Information

个 Label 控件,4 个 Edit 控件,3 个 Button 控件,1 个 ListBox 控件。其中 3 个 Label 控件的 Caption 属性分别为 To、CC、BCC、Subject Name、Attach Files;4 个 Edit 控件的 Text 属性都为空;3 个 Button 的 Caption 属性和 Name 属性分别为 Send Email/SendEmail、Add Files/AddFiles、Remove Files/RemoveFiles,设计结果如图 1.15 所示。

在【控件】工具栏的 Standard 选项卡中选中 Memo 控件,将其放在 TabSheet1 上,其缺

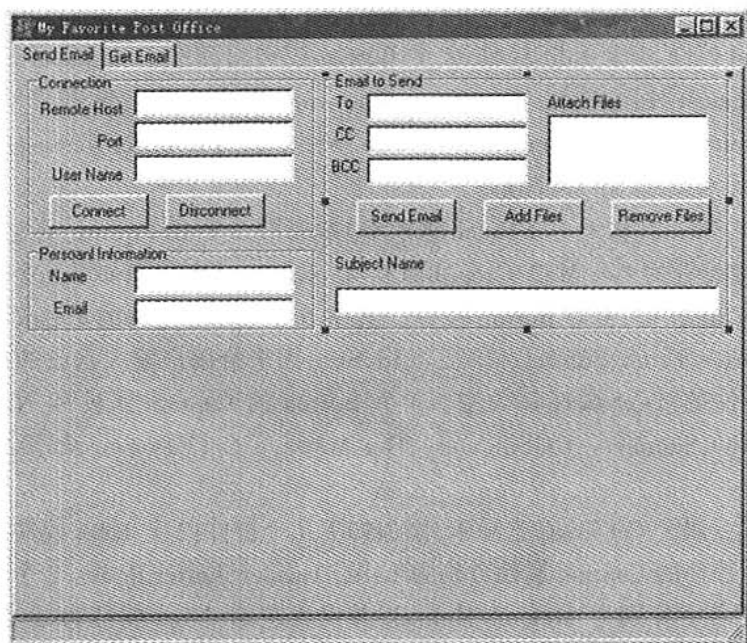


图 1.15 设计 Email to Send

省名字叫做 Memo1,并设置它的 Align 属性为 alClient,ScrollBar 属性为 ssVertical,清空 Lines 属性中的文本,并将【控件】工具栏的 FastNet 选项卡中的 NMSMTP 控件和【控件】工具栏的 Dialog 选项卡中的 OpenDialog 控件放在 Memo1 上。如图 1.16 所示,基本完成了 Send Email 选项卡的设计。

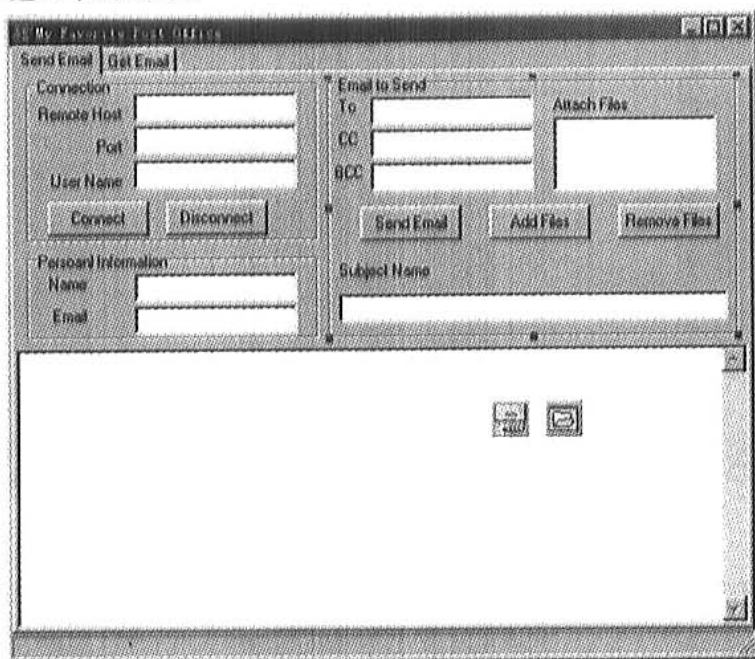


图 1.16 TabSheet1 的设计结果

选中 TabSheet2(属于 Get Email 选项卡),在里面添加如下一个 GroupBox,将它的 Caption 属性和 Name 属性设置为 Connection/GroupBox2。在 Connection 中添加如下几个控件:4 个 Label 控件,4 个 Edit 控件,2 个 Button 控件。其中 3 个 Label 控件的 Caption 属性分别为:Remote Host、Port、User Name、Password;其中 4 个 Edit 控件的 Text 属性都为空;2 个 Button 控件的 Caption 属性和 Name 属性分别为 Connect/Connect2、Disconnect/Disconnect2,设计结果如图 1.17 所示。

接着在 TabSheet2(属于 Get Email 选项卡)添加第二个 GroupBox,将它的 Caption 属性和 Name 属性设置为 Get Mail/GroupBox3。在 Get Mail 中添加如下几个控件:5 个 Label 控件,5 个 Edit 控件,3 个 Button 控件。其中 5 个 Label 控件的 Caption 属性分别为 Message to Retrieve、From、Subject、Message Id、Size,用来给用户输入自己的地址信息。其中 5 个 Edit 控件的 Text 属性都为空。3 个 Button 的 Caption 属性和 Name 属性分别为 Delete/Delete、Get Summary/GetSummary、Get Message/GetMessage,设计结果如图 1.18 所示。

在 TabSheet2(属于 Get Email 选项卡)添加如下几个控件:2 个 Label 控件,2 个 Memo 控件。2 个 Label 控件的 Caption 属性分别为 Letter Header、Letter Body。2 个 Memo 的属性设置如下:缺省名字分别为 Memo2/Memo3,并设置它的 Align 属性为 alClient,ScrollBar 属性为 ssVertical,清空 Lines 属性中的文本,如图 1.19 所示。

在【控件】工具栏的 FastNet 选项卡中选 NMPop3 控件,将其放在 Memo3 上,其缺

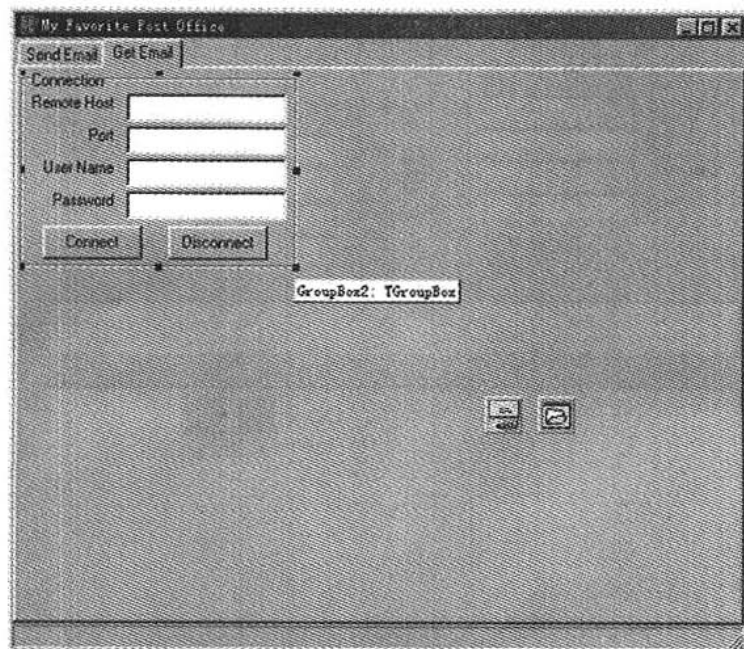


图 1.17 设计 Connection

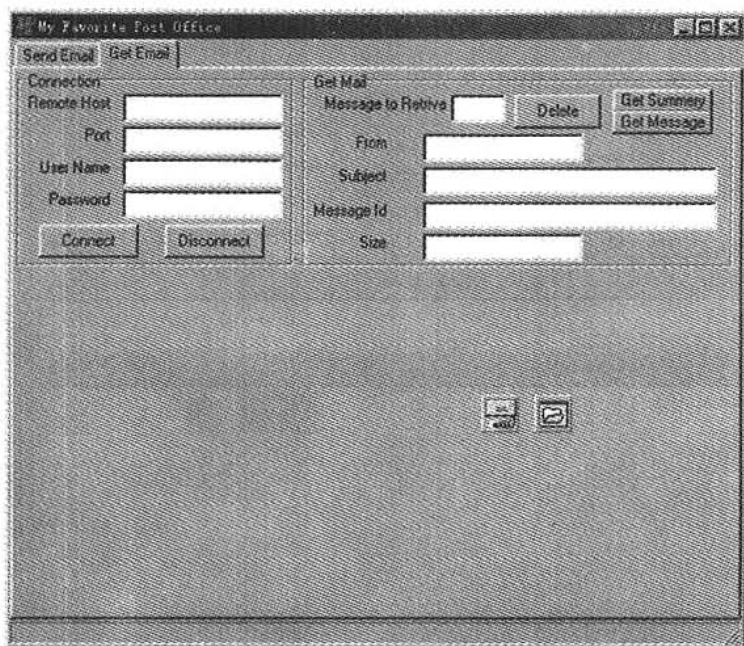


图 1.18 设计 Get Email

省名字叫做 NMPop3,如图 1.20 所示。

1.1.3 程序编制

下面,我们给上述设计的界面添加程序。

选择 Form1,双击其 OnCloseQuery 事件的右侧,生成缺省的 FormCloseQuery 事件,

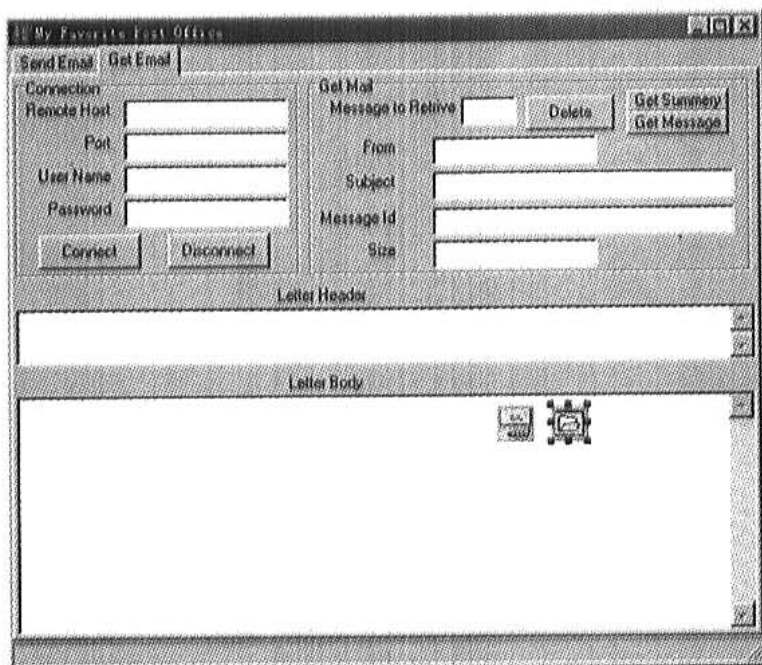


图 1.19 设计 TabSheet2

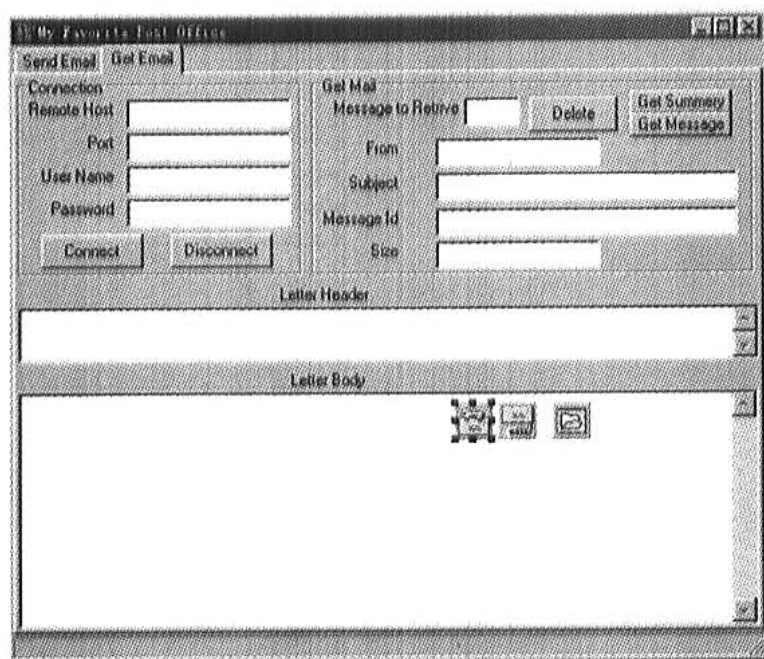


图 1.20 添加 NMPop3

添加处理代码如下：

```

procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    {添加代码开始处}
    {关闭窗体时,同时关闭 NMSMT}

```



```

NMSMTP1.Abort;
{关闭窗口时,同时关闭 NMPOP}
NMPOP31.Abort;
{添加代码结束处}

```

```
end;
```

进入 Send Email 选项卡中,选择 NMSMTP1,双击其 OnConnect 事件的右侧,生成缺省的 NMSMTP1Connect 事件,添加处理代码如下:

```

procedure TForm1.NMSMTP1Connect(Sender: TObject);
begin
    {添加代码开始处}
    StatusBar1.SimpleText := 'Connected';
    {添加代码结束处}

```

```
end;
```

选择 NMSMTP1,双击其 OnConnectionFailed 事件的右侧,生成缺省的 NMSMTP1ConnectionFailed 事件,添加处理代码如下:

```

procedure TForm1.NMSMTP1ConnectionFailed(Sender: TObject);
begin
    {添加代码开始处}
    ShowMessage('Connection Failed');
    {添加代码结束处}

```

```
end;
```

选择 NMSMTP1,双击其 OnConnectionRequired 事件的右侧,生成缺省的 NMSMTP1ConnectionRequired 事件,添加处理代码如下:

```

procedure TForm1.NMSMTP1ConnectionRequired(var Handled: Boolean);
begin
    {添加代码开始处}
    {询问是否进行连接}
    If MessageDlg('Connection Required. Connect?', mtConfirmation, mbOkCancel, 0)
        = mrOk then
    begin
        Handled := TRUE;
        {如果单击确认按钮,则进行连接}
        NMSMTP1.Connect;
    end;
    {添加代码结束处}

```

```
end;
```

选择 NMSMTP1,双击其 OnDisconnect 事件的右侧,生成缺省的 NMSMTP1Disconnect 事件,添加处理代码如下:

```

procedure TForm1.NMSMTP1Disconnect(Sender: TObject);

```

```
begin
    {添加代码开始处}
    If StatusBar1 < > nil then
    {当断开连接,则显示“Disconnected”信息}
        StatusBar1.SimpleText := 'Disconnected';
    {添加代码结束处}
end;
```

选择 NMSMTP1, 双击其 OnEncodeEnd 事件的右侧, 生成缺省的 NMSMTP1EncodeEnd 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1EncodeEnd(Filename: String);
begin
    {添加代码开始处}
    StatusBar1.SimpleText := 'Finished encoding ' + Filename;
    {添加代码结束处}
end;
```

选择 NMSMTP1, 双击其 OnEncodeStart 事件的右侧, 生成缺省的 NMSMTP1EncodeStart 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1EncodeStart(Filename: String);
begin
    {添加代码开始处}
    StatusBar1.SimpleText := 'Encoding ' + Filename;
    {添加代码结束处}
end;
```

选择 NMSMTP1, 双击其 OnFailure 事件的右侧, 生成缺省的 NMSMTP1Failure 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1Failure(Sender: TObject);
begin
    {添加代码开始处}
    StatusBar1.SimpleText := 'Failure';
    {添加代码结束处}
end;
```

选择 NMSMTP1, 双击其 OnHeaderIncomplete 事件的右侧, 生成缺省的 NMSMTP1HeaderIncomplete 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1HeaderIncomplete(var handled: Boolean;
    hiType: Integer);
begin
    {添加代码开始处}
    ShowMessage('Header Incomplete. ');
    {添加代码结束处}
```

end;

选择 NMSMTP1, 双击其 OnHostResolved 事件的右侧, 生成缺省的 NMSMTP1HostResolved 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1HostResolved(Sender: TComponent);
begin
    {添加代码开始处}
    StatusBar1.SimpleText := 'Host Resolved';
    {添加代码结束处}
```

end;

选择 NMSMTP1, 双击其 OnInvalidHost 事件的右侧, 生成缺省的 NMSMTP1InvalidHost 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1InvalidHost(var Handled: Boolean);
    {添加代码开始处}
var
    TmpStr: String;
begin
    {当远程主机地址不能连接时要求输入另外的一个地址}
    If InputQuery('Invalid Host!', 'Specify a new host:', TmpStr) then
        begin
            {远程主机地址为用户键入的新地址}
            NMSMTP1.Host := TmpStr;
            Handled := TRUE;
        end;
    {添加代码结束处}
```

end;

选择 NMSMTP1, 双击其 OnMailListReturn 事件的右侧, 生成缺省的 NMSMTP1MailListReturn 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1MailListReturn(MailAddress: String);
begin
    {添加代码开始处}
    Memo2.Lines.Add(MailAddress);
    {添加代码结束处}
```

end;

选择 NMSMTP1, 双击其 OnPacketSent 事件的右侧, 生成缺省的 NMSMTP1PacketSent 事件, 添加处理代码如下:

```
procedure TForm1.NMSMTP1PacketSent(Sender: TObject);
begin
    {添加代码开始处}
    StatusBar1.SimpleText := IntToStr(NMSMTP1.BytesSent) + 'bytes of ' + IntToStr
```