

计算机编程技巧

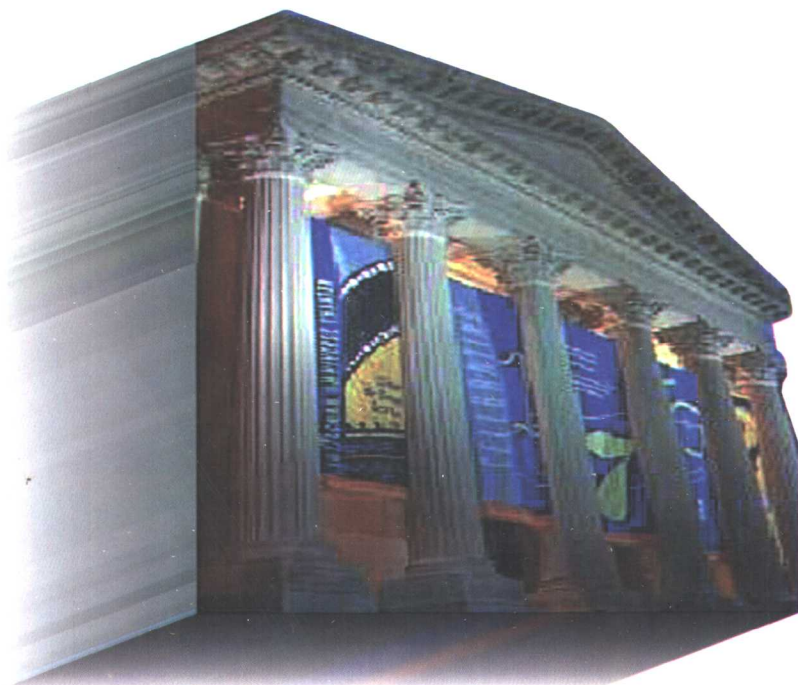
ABC

系列丛书

Delphi

编程技巧

网络与数据库篇



清宏计算机工作室
编著



机械工业出版社
China Machine Press

计算机编程技巧 ABC 系列丛书

Delphi 编程技巧

(网络与数据库篇)

清宏计算机工作室 编著



机械工业出版社

本书主要介绍用 Delphi 进行网络和数据库编程的一些技巧,从基本技巧到高级技巧都有介绍。网络部分从基本的网络控件的使用技巧讲起,再讲到网络 API 的使用和在程序中调用 DOS Shell 命令的技巧;DOS Shell 的引入会使一些非常棘手的问题变得非常容易;网络部分最后介绍了一些其他技巧,如 Cookie 的使用和 Web Server 的建立等。数据库部分介绍了一些常用的数据库使用技巧,BDE 的一些 API 函数的使用。

由于本书讲解的是技巧,所以读者最好对 Delphi 有一定程度的掌握和编程经历,但这个要求也不是必需,只要读者熟悉了 Object Pascal 的语法,书中的程序一般能看懂(不懂的地方可以看联机帮助)。相信通过此书可以使读者的网络和数据库编程的水平更上一层楼。

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:边 萌 封面设计:姚 毅

责任印制:郭景龙

三河市宏达印刷厂印刷·新华书店北京发行所发行

2001 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·22.5 印张·555 千字

0 001—5000 册

定价:40.00 元 (1CD,含配套书)

ISBN7-900043-46-2/TP·42

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68993821、68326677-2527

前 言

随着计算机技术的飞速发展，计算机的开发迅速得到普及推广。当前有一大批计算机开发人员，虽然已经掌握了某种开发工具的基础知识、甚至达到了相当高的水平，但仍有很多盲区，或者急需更多的编程经验和技巧。为此，我们决定组织编写这套“计算机编程技巧 ABC 系列丛书”，以满足广大读者的需求，帮助读者快速成为计算机编程得行家里手。

这套丛书针对当前最流行的几种开发语言，介绍开发过程中的编程经验和技巧，并解决开发过程中的疑点和难点，主要突出“技巧”二字。针对某些开发语言，根据其内容相关性和独立性，我们编写了两个系列的书籍，即“网络与数据库篇”和“多媒体与系统篇”。这样可以使读者有更大的选择余地，并且在内容编排、读者阅读上都更为合理。

每本书按章节编写，各章由既相对独立，又相互关联的技巧主题组成，并且在顺序编排上考虑了内容的前后连续性。

针对每一个技巧主题，我们将其分为“ABC”三个部分进行讲解。

A——关键所在（或原理方法） 以简单明了的语言指出实现技巧的关键所在或原理方法。

B——实现与应用 以一个简单、完整实例向读者介绍技巧的实现方法或步骤。

C——专家点评 对技巧加以适当说明，包括补充解释，强调注意事项等。比如简要介绍实现该技巧的其他方法，或者介绍关键技术的扩展应用等。

针对每一个技巧主题，力求在内容讲解清楚的前提下，篇幅短小，使读者在有限篇幅内学到更多的技巧，整套书充分体现强调短小精悍、高效率的特点。

读者在阅读本套丛书之前，应该对相应开发语言有了一定了解，但并不要求读者开发水平一定很高，在章节编排和内容讲解过程中，我们都尽量照顾了水平还比较低的读者，因此初学者还是可以阅读这套丛书的。

希望这套丛书能使你很快成为软件开发的高手。

恳请广大读者对书中不足之处提出宝贵意见。

清宏计算机工作室

编者的话

记得我最开始学习 Delphi 时，原因很简单，就是看中了面板上的那一堆控件。VC 的 MFC 难以做出美观的界面，VB 的语法也太松散，都是我不喜欢的。随着学习的深入和实践经验的不断增加，我越来越感到 Delphi 的魅力，其博大精深是 VB 无法媲美的，和 VC 相比也不至于太逊色。作为流行的 RAD 工具之一，Delphi（还有 Inprise 公司的另一个工具 C++ Builder）真正体现了使用简便、开发快速的特点，而如果用户需要做非常底层的应用（如图形处理算法和直接对硬件端口的操作），那么 Delphi 也同样是一个好选择，它几乎对 Windows 所有的 API 都进行了封装，opengl 单元中还提供大量的涉及到 OpenGL 编程的函数，用户也可以直接在程序中嵌入汇编代码段，一方面可以使程序直接操作硬件，另一方面，即使不操作硬件，汇编的使用也使程序代码的执行效率大为提高。如果既需要美观的界面，又需要底层的操作，那么 Delphi（或 BCB）将是您最佳的选择。关于上面讨论到的三个开发工具的比较，很多文章都做了专门的介绍，在此就不再重复，最后只想说明，Delphi 的队伍在不断扩大，无论在学术性强的学校还是在面向商业的公司（据我所知北大方正、中软还有其他的一些小软件公司的 Windows 主要编程工具就是 Delphi 和 BCB），Delphi 都拥有众多的用户，而且必将越来越多，“真正的程序员用 C（将来或许就是 C# 了），聪明的程序员用 Delphi”，看来世界上聪明的程序员还是越来越多，当然所有这些都要归功于 Inprise（原 Borland）公司对用户的真正体贴和付出的辛勤的汗水，让我们向他们表示由衷的感谢！

本书是《计算机编程技巧 ABC 系列丛书》之一，取材限于网络和数据库，作者在这方面有实际的开发经验，讲解内容力求全而精。但由于作者水平有限，再加上时间紧迫，虽然经常通宵工作，但疏漏或不妥之处难免，如有什么疑问，欢迎读者对出现的问题进行咨询，当然如果读者有实现某些功能的别的方法或更好的途径，也欢迎来信探讨。

本书分为两个部分：网络和数据库。共分 7 章讲解，第 1~5 章讲解网络的编程技巧，第 6~7 章讲解数据库的编程技巧。

第1章 用 Delphi 自带组件进行网络编程的基本技巧。主要介绍与网络编程有关的 Delphi 控件的基本使用技巧，例如如何使用 TNMFTP 组件实现从 ftp 服务器上下载文件，如何使用 TNMPOP3 组件和 TNMSMTP 组件实现收发电子邮件等。有些实例后面的“专家说明”部分对涉及的一些协议也做了深入的介绍，以便对这方面陌生的读者对该组件的使用有更深入的认识。当然，熟悉的读者可以跳过这些介绍。

第2章 用 Delphi 自带组件进行网络编程的高级技巧。主要讲解与网络编程有关的 Delphi 控件的高级使用技巧，此部分在第 1 章的基础上更深入地介绍了网络组件的使用方法和诀窍，如使用 TNMUDP 组件实现远程控制等。

第3章 用 API 函数深入网络编程。深入介绍网络 API 详细的使用方法，包括 MSDN 的原始函数定义、功能说明、各参数的类型和含义等。此部分是本书的特色之一。

第4章 调用 DOS Shell 命令。在 Windows 程序中调用 DOS Shell 命令来为我们的编程开辟蹊径。主要介绍了 DOS 下常用的与网络有关的命令的功能，所带的开关选项的作用，如

何实现用 WINEXEC 这个 API 函数来调用这些命令以及如何消除写临时文件的延迟所造成的每次命令需要执行两次才能显示正确结果等，对网络的 DOS 命令不熟悉或者对 Windows 程序中调用 DOS Shell 命令有疑问的读者不妨参考。

第5章 网络编程其他技巧。不能归入第 1~4 章中的一些网络编程技巧都并入此章中。

第6章 使用数据库的常用技巧。主要介绍对使用数据库经常出现的问题的解决办法以及基本数据库控件的使用技巧。

第7章 使用 BDE 的 API 函数获取数据库系统信息。深入介绍 BDE 的 API 函数的使用方法，包括函数原型、函数作用、各参数的类型和含义等。这是本书的特色之二。

在每一节中都有范例的程序清单。这是为了使用户更加清楚程序的结构和源代码，对一些特别重要的地方，专门拿出来做了说明。作者已尽可能使介绍详细而又简洁，范例程序尽可能地简短，当然为了更详细说明，有些范例略显不够简洁。鱼和熊掌不可兼得，作者只能选其一，如果读者觉得有些章节冗长，还敬请谅解。

本书的配套光盘

书中出现的实例程序源代码保存在本书的配套光盘中，主要是避免读者测试程序时繁琐的手工录入。这虽然方便了读者，但也正如硬币都有正反两面一样，很可能同时也使读者对代码的熟悉程度和技巧的领会程度降低。但无论如何，将代码附上是我们对读者尽责，至于如何使用就需要读者根据个人自己的具体情况来选择了。为了帮助读者更好地领会这些技巧，作者的建议是：先看一遍代码，抓住每个实例中的关键技巧，然后运行一下光盘中相应的实例程序，看看程序具体的运行结果，如果读者觉得已经掌握，那么强烈建议读者自己设计类似的程序来检验自己的想法，或者对实例程序进行修改或改进。

与作者的联系方式

hellins@263.net

此地址不会改变，当“263”不再提供邮件服务时除外。

由于时间关系和水平有限，书中难免出错，希望读者谅解并批评指正。

目 录

前言

编者的话

第 1 章 自带组件网络编程基本技巧	1
1.1 用 TNMDayTime 控件实现时间查询	1
1.2 用 TNMMsg、TNMMsgServ 组件收发简单 ASCII 码消息	5
1.3 用 TNMEcho 组件测试网络网速	12
1.4 用 TNMURL 组件对 URL 数据编码和解码	16
1.5 用 TNMUUProcessor 组件实现 MIME、UUEncode 编码解码	19
1.6 Delphi 中用 TWebBrowser 作 Html 格式帮助文件	26
1.7 用 TServerSocket、TClientSocket 组件编网络聊天室	32
1.8 用 TNMFinger 组件从 Finger 服务器得到用户信息	41
第 2 章 网 络 编 程	47
2.1 用 TNMUDP 组件实现远程控制	47
2.2 用 TNMHttp 实现域名到 IP 转换并获得自己 IP	69
2.3 用 TNMDayTime 控件得知自己 IP 地址	73
2.4 如何用 TNMSMTP 发送匿名邮件	75
2.5 如何在程序中加入网上的图片	80
第 3 章 用 API 函数深入网络编程	82
3.1 实现超链接	82
3.2 定制 THyperLink 组件处理超链接	85
3.3 得到本机机器名、IP 地址及其类别	90
3.4 得到本机 MAC 地址	94
3.5 得到和设置机器名	99
3.6 实现拨号连接	101
3.7 实现 Ping 操作	103
3.8 局域网中检测某台机器是否在网上	109
3.9 局域网中通过计算机名得到 IP	111
3.10 通过 IP 地址得到计算机名	114
3.11 在 WinNT/2000 网络邻居中获取所有的工作组（一）	117
3.12 在 WinNT/2000 网络邻居中获取所有的工作组（二）	123
3.13 在网络邻居中获取指定工作组内的所有计算机	127
3.14 在网络邻居中获取指定计算机的共享资源信息	131
3.15 在网络邻居上得到某台机器的磁盘空间	135
3.16 网络邻居中的文件拷贝	139

3.17	实现网络驱动器的映射和断开	141
3.18	NT/Win2000 中利用 API 实现 Winpop 发送消息	146
3.19	在自己的程序中关闭 IE 窗口	151
第 4 章	调用 DOS SHELL 命令为网络编程开辟蹊径	154
4.1	NT/Win2000 下得到 DNS 服务器域名以及 IP	154
4.2	NT/Win2000 中实现目录共享与断开共享	157
4.3	查看网络邻居中某台机器的共享资源	163
4.4	得到本机名、IP 地址、MAC 地址、子网掩码、默认网关及网卡品牌	167
4.5	得到局域网上某台机器的日期和时间	170
4.6	得到网络邻居中指定计算机所属工作组及其 MAC 地址	174
4.7	调用 hostname 命令得到本机名	177
4.8	获取路由信息、默认网关、以及网卡信息	179
4.9	获取与远程机连接时所经过的路由信息	182
4.10	列举本工作组中所有计算机	185
第 5 章	网络编程其他技巧	188
5.1	Cookie 的建立和使用	188
5.2	Web Server 程序响应用户的输入	190
5.3	Web Server 如何返回图像	194
5.4	用 DDE 技术调用 NetScape 链接指定站点	197
5.5	如何获取 IE 的 URL 栏中的地址	200
5.6	用 API 实现串口通信	202
第 6 章	数据库使用常用技巧	207
6.1	Delphi 数据集过滤技巧	207
6.2	处理 word 文档与数据库的互联	208
6.3	处理数据库日期型字段的显示与输入	215
6.4	动态改变 DBGrid 的颜色	222
6.5	定制 BDE 驱动程序以精简 Delphi 数据库应用系统	225
6.6	解决打开一个 DBF 表时的 “Index not found.....” 错误	228
6.7	如何 Pack、Undelete dBase 或 FoxPro 数据表	230
6.8	如何在一个 Dbgrid 中显示多数据库	238
6.9	实现多库关联查询	241
6.10	使用 SQL 实现字段数据的模糊查询	244
6.11	通过注册表在程序中增减和修改数据源	247
6.12	图像数据的存取	251
6.13	用 Delphi 进行数据库之间转换	259
6.14	自动登陆数据库	264
6.15	自动检测、建立数据库别名和数据表	266
6.16	在程序中动态地建立和使用别名(一)	272

6.17	在程序中动态地建立和使用别名(二)	274
6.18	在程序中动态地建立和使用别名(三)	278
6.19	在程序中动态地建立和使用别名(四)	281
6.20	用 Delphi 开发 ASP 文件上传组件	283
6.21	开发 WEB MAIL 程序	290
第 7 章	使用 BDE 的 API 函数获取数据库系统信息	301
7.1	获取 BDE 的配置信息	301
7.2	获取 BDE 系统的版本信息	304
7.3	获取 BDE 系统的状态信息	307
7.4	获取客户端或应用程序的系统级信息	310
7.5	获取系统可用的数据库和别名列表	313
7.6	获取合法驱动列表并加载指定的驱动	316
7.7	获取指定驱动の詳細信息	320
7.8	获取指定驱动支持的索引名	324
7.9	获取指定类型索引的详细信息	327
7.10	获取指定驱动所支持的表的类别	332
7.11	获取指定类别数据表的性能的信息	335
7.12	获取指定类型表支持的字段	339
7.13	获取数据表字段的详细信息	342
7.14	取得 BDE 中的 Alias 所指向的数据库的实际目录	347

第 1 章 自带组件网络编程基本技巧

1.1 用 TNMDayTime 控件实现时间查询



原理方法

Delphi 的 FastNet 面板中有 TNMDayTime 控件，作用是用来查询远程机时间服务器的日期和时间的（当然并不是任何计算机都提供此项服务，这需要时间服务器的支持）。只要设置好 TNMDayTime 控件的 Host（远程主机 IP 地址，一些主要的时间服务器后面有附表）和 Port（时间服务器的端口，一般是 13），通过 TNMDayTime 控件的 DayTimeStr 属性即可方便地得知远程机的日期和时间。



实现与应用

新建一应用程序，加入如下控件：

```
NMDayTime1: TNMDayTime;  
Button1: TButton;  
Edit1: TEdit;           //供用户输入日期时间服务器 IP 地址  
Edit2: TEdit;           //供用户输入日期时间服务器端口  
StaticText1: TStaticText;  
StaticText2: TStaticText;  
StaticText3: TStaticText;  
Memo1: TMemo;          //显示查询结果  
Bevel1: TBevel;
```

其中一些控件属性与默认值的不同如下：

```
Edit1:  
    Text = '202.112.1.58' //这是一个提供日期时间服务的服务器  
Edit2:  
    Text = '13'  
Memo1:  
    ReadOnly = True      //设为只读  
    ScrollBars = ssVertical //要竖直滚动条  
NMDayTime1:  
    Host = '202.112.0.70' //这也是一个时间服务器的地址，如果用户不输入则用
```

此默认值

```
Port = 13
```

```
TimeOut = 5000 //超时限制为 5 秒，此属性最好设置大些，主要视网速决定
```

```
ReportLevel = 0 //此属性控制 OnStatus 时间和 Status 属性所报告的细节的多  
少，有如下 6 个取值：
```

```
Status_None = 0
```

```
Status_Informational = 1
```

```
Status_Basic = 2
```

```
Status_Routines = 4
```

```
Status_Debug = 8
```

```
Status_Trace = 16
```

级别设置越高，则得到的细节越多，作为时间服务器，我们只通过 DayTimeStr 属性得到日期时间信息，并不使用 Status 属性，所以此属性设为 0 最节省网络资源。

程序清单如下：

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
```

```
ExtCtrls, StdCtrls, Psock, NMDayTim;
```

```
//Psock, NMDayTim 两个单元是加入 TdayTime 控件时 Delphi 自动加入的
```

```
type
```

```
TForm1 = class(TForm)
```

```
  NMDayTime1: TNMDayTime;
```

```
  Button1: TButton;
```

```
  Edit1: TEdit;
```

```
  Edit2: TEdit;
```

```
  StaticText1: TStaticText;
```

```
  StaticText2: TStaticText;
```

```
  StaticText3: TStaticText;
```

```
  Memo1: TMemo;
```

```
  Bevel1: TBevel;
```

```
  procedure Button1Click(Sender: TObject);
```

```
  procedure NMDayTime1InvalidHost(var Handled: Boolean);
```

```
  procedure FormCreate(Sender: TObject);
```

```
private
```

```
{ Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);

var

temp:string;
begin
  try
    NMDayTime1.host:=edit1.text;
    NMDayTime1.port:=strtoint(edit2.text);
    //此处不要画蛇添足地加上 NMDayTime1.Connect, 因为设置好 Host 和 Port 属性后,
    //NMDayTime1 已自动连接服务器, 加上后程序运行时会报错: 已经连接!
    temp:=trim(NMDayTime1.DayTimeStr); //去掉返回的日期时间 DayTimeStr 结尾的回
    车, //使 Memo1 中的显示更加紧凑。
    memo1.lines.add(temp);
  except
    raise exception.create('服务器不能连接!');
    //不能连接服务器则引发异常
  end;
  NMDayTime1.Disconnect;
  //每次查询后要断开连接。
end;

procedure TForm1.NMDayTime1InvalidHost(var Handled: Boolean);
begin
  showmessage('无效服务器! ');
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
    memo1.lines.Clear;  
    if NMDayTime1.Connected then  
        NMDayTime1.DisConnect;  
    //如果 NMDayTime1 处于连接状态则断开  
end;  
  
end.
```

程序运行结果如图 1-1 所示。

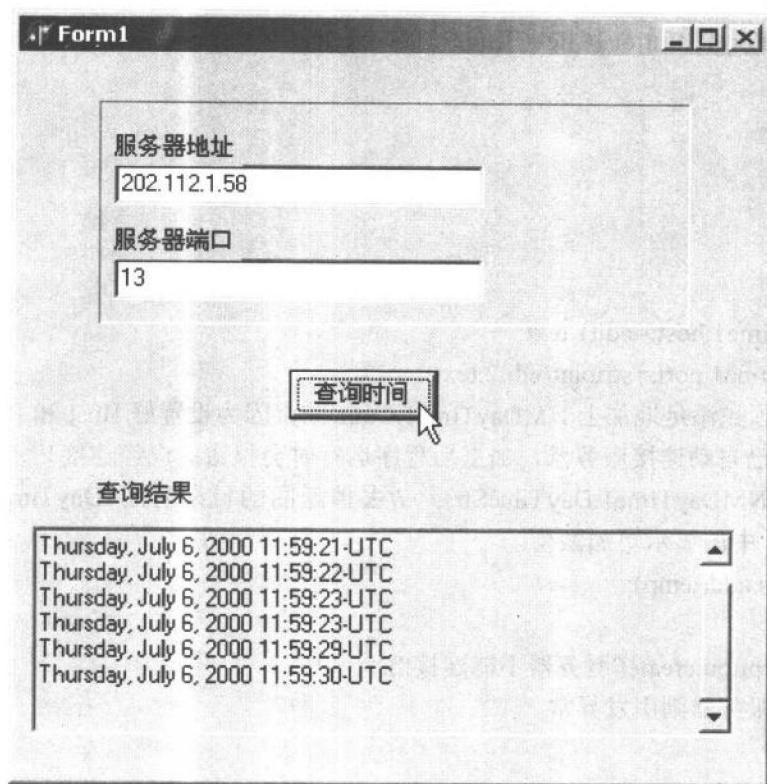


图 1-1 用 TdayTime 控件实现时间查询运行结果



专家点评

- (1) 可以改进程序, 用 TdayTime 的 Proxy 和 ProxyPort 属性指定代理服务器地址和端口。
- (2) 表 1-1 给出了一些日期时间服务器, 这样的服务器比较多。

表 1-1 国内 202.112 网段一些日期时间服务器

IP 地 址	端 口
202.112.0.70	13
202.112.1.58	13
202.112.1.74	13
202.112.1.204	13
202.112.1.220	13
202.112.1.251	13
202.112.1.252	13
202.112.4.61	13
202.112.4.65	13
202.112.4.66	13
202.112.4.68	13
202.112.4.101	13
202.112.4.122	13
202.112.4.124	13
202.112.4.129	13
202.112.5.33	13
202.112.5.34	13
202.112.5.62	13
202.112.5.118	13
202.112.5.180	13
202.112.6.194	13
202.112.6.205	13
202.112.7.1	13
202.112.7.2	13
202.112.7.6	13
202.112.7.11	13
202.112.7.12	13
202.112.7.14	13
202.112.7.15	13
202.112.10.2	13
202.112.10.95	13
202.112.11.130	13
202.112.11.132	13
202.112.11.136	13
202.112.11.208	13
202.112.11.211	13
202.112.13.42	13

1.2 用 TNMMsg、TNMMsgServ 组件收发简单 ASCII 码消息



原理方法

TNMMsg 是用来在局域网和广域网上发送简单的 ASCII 码文本消息的组件,基于 TCP/IP 协议,一台计算机上(可称其为主机或服务器)用 TNMMsgServ 或 TNMMsgGeneralServer 创建程序监听本机的某一个端口,那么另一台机器(可成为客户机)上的用 TNMMsg 组件创建的程序就可以通过此端口往本机发送消息。



实现与应用

新建一应用程序, 加入如下控件:

```
Panel1: TPanel;  
Memo1: TMemo;           //显示对方消息  
Edit1: TEdit;           //输入要发送的消息  
Edit2: TEdit;           //指定自己代号  
Edit3: TEdit;           //指定对方地址, 可以为 IP 地址或计算机名  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
NMMsg1: TNMMsg;  
NMMSGServ1: TNMMSGServ;  
StatusBar1: TStatusBar;
```

主要属性与默认值不同之处如下:

```
object Form1: TForm1  
  Caption = 'Msg Demo'  
object Panel1: TPanel  
  Align = alBottom        //窗体大小改变时能够自动跟着改变大小  
  OnResize = Panel1Resize  
end  
object Label1: TLabel  
  Caption = '你的代号'  
  Font.Name = '宋体'  
end  
  object Label2: TLabel  
    Caption = '主机'  
    Font.Name = '宋体'  
  end  
object Label3: TLabel  
  Caption = '输入信息并按回车键'  
  Font.Name = '宋体'  
end  
object Edit1: TEdit  
  OnKeyPress = Edit1KeyPress  
end  
object Edit2: TEdit  
  Text = 'hellins'  
end  
object Edit3: TEdit  
  Text = '127.0.0.1'      //本机 LoopBack 地址, 即向本机发送消息
```

```

    end
  end
  object Memo1: TMemo
    Align = alClient    //窗体大小改变时能够自动跟着改变大小
  end
  object StatusBar1: TStatusBar
    SimplePanel = True  //状态条能显示消息
  end
  object NMMsg1: TNMMsg
    Port = 6711          //自定义端口，只要不冲突，可以为别的值
    OnDisconnect = NMMsg1Disconnect
    OnConnect = NMMsg1Connect
    OnInvalidHost = NMMsg1InvalidHost
    OnHostResolved = NMMsg1HostResolved
    OnStatus = NMMsg1Status
    OnConnectionFailed = NMMsg1ConnectionFailed
    OnMessageSent = NMMsg1MessageSent
  end
  object NMMSGServ1: TNMMSGServ
    Port = 6711
    OnStatus = NMMSGServ1Status
    OnClientContact = NMMSGServ1ClientContact
    OnMSG = NMMSGServ1MSG
  end
end

```

程序清单及其详细解释如下：

```
unit msgdem;
```

```
interface
```

```
uses
```

```
  Windows, SysUtils, Classes, Controls, Forms, Dialogs,
```

```
  StdCtrls, ExtCtrls, NMMSG, ScktComp, ComCtrls, Psock;
```

```
  // NMMSG 为控件当加入 TNMMSG、TNMMSGSERV 后 Delphi 自动加入
```

```
type
```

```
  TForm1 = class(TForm)
```

```
    Panel1: TPanel;
```

```
    Memo1: TMemo;
```

```
    Edit1: TEdit;
```

```
NMMsg1: TNMMsg;  
NMMSGServ1: TNMMSGServ;  
Label1: TLabel;  
Edit2: TEdit;  
Label2: TLabel;  
Edit3: TEdit;  
StatusBar1: TStatusBar;  
Label3: TLabel;  
procedure Panel1Resize(Sender: TObject);  
procedure Edit1KeyPress(Sender: TObject; var Key: Char);  
procedure NMMSGServ1MSG(Sender: TComponent; const sFrom, sMsg: String);  
procedure NMMsg1MessageSent(Sender: TObject);  
procedure NMMsg1Connect(Sender: TObject);  
procedure NMMsg1ConnectionFailed(Sender: TObject);  
procedure NMMsg1Disconnect(Sender: TObject);  
procedure NMMsg1HostResolved(Sender: TComponent);  
procedure NMMsg1Status(Sender: TComponent; Status: String);  
procedure NMMsg1InvalidHost(var handled: Boolean);  
procedure NMMSGServ1ClientContact(Sender: TObject);  
procedure NMMSGServ1Status(Sender: TComponent; Status: String);  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;  
  
var  
    Form1: TForm1;  
  
implementation  
  
{$R *.DFM}  
  
procedure TForm1.Panel1Resize(Sender: TObject);  
begin  
    Edit1.Left := 8;  
    Edit1.Width := Panel1.Width - 16;  
    // 上面两句使 Edit1 的长度随面板 Panel1 的改变而自动适应  
    Label1.Left := 8;
```