

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列

编程新概念丛书 4

Visual Basic 6

网络编程实例教程

北京希望电子出版社 总策划

肖斌 董磊 编写



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列

编程新概念丛书 4

Visual Basic 6

网络编程实例教程

北京希望电子出版社 总策划

肖斌等 编写



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书是关于 Visual Basic 6 网络编程的专著。本书从实用的角度出发,用 **50** 个实例详细讲述了使用 Visual Basic 6 进行网络编程的方法与技巧。全书由 13 章构成,主要内容包括:获取网络配置信息、获取网卡物理地址、获取工作组及用户信息、获取工作站配置与用户信息、获取网络机器的服务信息、浏览器编程、获取网络连接信息、获取网络资源信息、TCP/IP 协议编程、获取服务器信息、Ping 与路由追踪、DCHP 编程、URL 编程;同时在书尾给出了附录 A,详细介绍了调用 Windows API 的注意事项。

本书文字简洁,层次清晰,基础知识与具体实例相结合,例子翔实丰富,具有很强的实用性和指导性。本书既适合高校相关专业师生教学、自学用书,同时也可作为专业程序员自学指导书,社会 Visual Basic 培训班的参考书。

本书实例的源代码请读者在 www.b-xr.com 上去自由下载。

读者在使用本书过程中的技术问题,请与 zhusb@263.net 联系。

系 列 书 名 : “十五”国家重点电子出版物规划项目 计算机知识普及和软件开发系列
编程新概念丛书(4)

书 名 : Visual Basic 6网络编程实例教程

总 策 划 : 北京希望电子出版社

文 本 著 作 者 : 肖斌、董磊

责 任 编 辑 : 赵 汶

出版、发行者 : 北京希望电子出版社

地 址 : 北京市海淀区知春路63号卫星大厦三层 100080

网址: www.bhp.com.cn

E-mail: lwm@bhp.com.cn

电 话 : 010-62520290,62521724,62528991,62630301,62524940,62521921,82610344

(发行) 010-82675588-202(门市) 010-82675588-501,82675588-201(编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 希望图书输出中心 张月岭

文 本 印 刷 者 : 北京东升印刷厂

开本 / 规格 : 787 毫米×1092 毫米 16 开本 21.75 印张 514 千字

版次 / 印次 : 2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

印 数 : 1-3000 册

本 版 号 : ISBN 7-900118-06-3

定 价 : 33.00 元

说明: 凡我社产品如有残缺,可执相关凭证与本社调换。

前言

Visual Basic 软件具有易学习、功能强、见效快等特点，因而倍受人们的青睐。它不仅
是专业人员得心应手的开发工具，而且容易被非专业人员所掌握。

随着 Visual Basic 版本的更新和功能的增强，其应用范围也越来越广。对于一般的应
用程序而言，Visual Basic 可以完全胜任；但对于一些较为深入的应用，Visual Basic 则可能
会显得力不从心。此时，使用功能强大的 Windows API 函数是一种很不错的方案，灵活地
运用 Windows API 将使用户的程序更专业，功能更强大。

Windows 系统提供了大量的功能强大的 API 函数，利用这些 API 函数，可以编制出界
面友好、功能强大的应用程序，很好地扩充和延伸 Visual Basic 的功能，更充分地发挥
Windows 系统的性能。

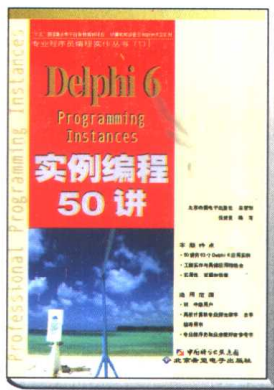
因此，如果要开发出更灵活、更实用、更具效率的应用程序，必然会涉及到直接使用
API 函数。虽然类库和控件使应用程序的开发简单了许多，但它们只提供 Windows 的一般
功能，对于比较复杂和特殊的功能来说，使用类库和控件是非常难以实现的，这时就需要采
用 API 函数来实现。

基于此，作者编写了《Visual Basic 6 网络编程实例教程》一书。通过对 Windows API
函数的分类，结合一些非常实用、有趣的实例，深入讲解 API 函数的应用，能使读者快速、
深入地掌握 Windows API 编程的方法与技巧。全书共分为 13 章，从多个方面探讨了 Windows
网络 API 的编程应用技术。本书的最大特点是：循序渐进，层次清晰，基础知识与具体实
例相结合，例子翔实丰富，具有很强的实用性和指导性。书中所有实例都是作者精心安排的，
操作性极强，本书既适合高校相关专业师生教学、自学用书，同时也可作为专业程序员自学
指导书，社会 Visual Basic 培训班的参考书。

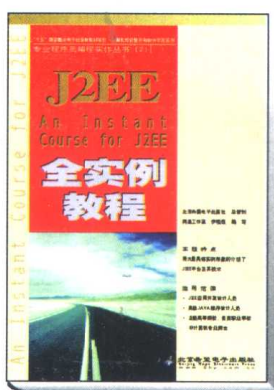
本书由肖斌、董磊、廖育荣、李启军、汪荣峰、王俊辉等编写，另外参加编写工作和
提供技术帮助的还有：庄剑南、江亚萍、牛静、张向龙、胡敏、任玮、张超等，在此对他们
为本书所付出的辛勤劳动一并表示感谢！

由于时间仓促，再加上作者水平有限，书中难免会有一些缺陷和不足之处，希望广大
读者不吝赐教并提出宝贵意见。

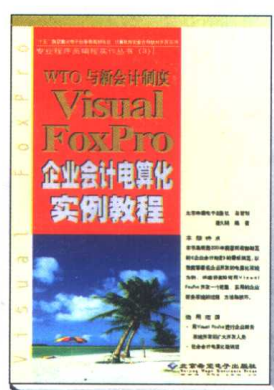
编程的利器、知识的进



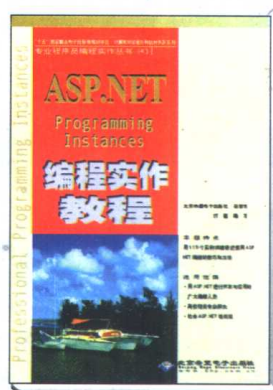
CX-3725
定价: 39.00 元
ISBN: 7900088962



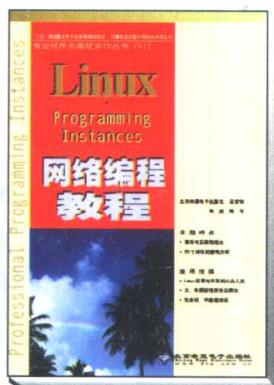
CX-3770
定价: 55.00 元
ISBN: 7900101330



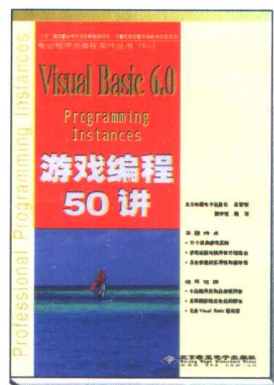
CX-3763
定价: 23.00 元
ISBN: 7900101136



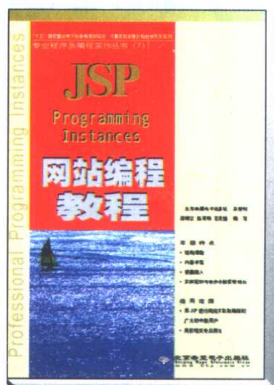
CX-3779
定价: 33.00 元
ISBN: 7900101179



CX-3786
定价: 52.00 元
ISBN: 7900101241



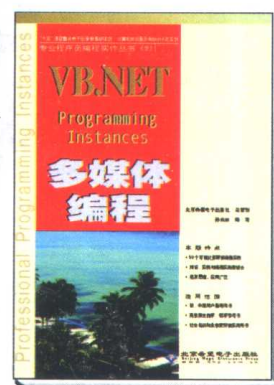
CX-3791
定价: 33.00 元
ISBN: 7900101292



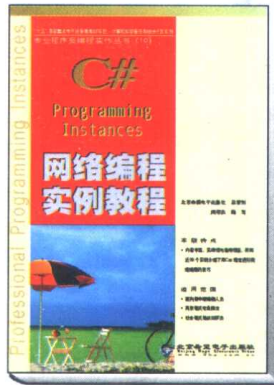
CX-3796
定价: 39.00 元
ISBN: 7900101322



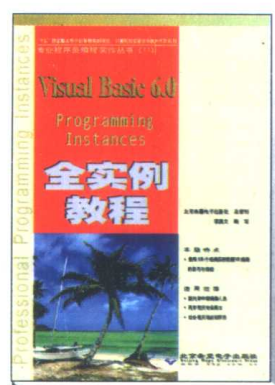
CX-3797
定价: 50.00 元
ISBN: 7900101314



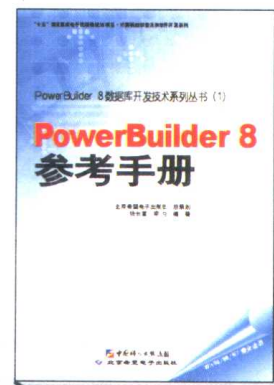
CX-3830
定价: 48.00 元
ISBN: 7900101357



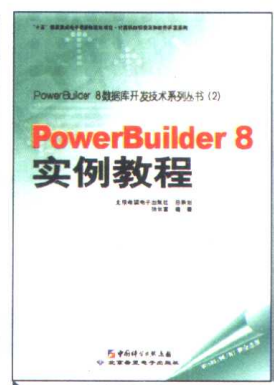
CX-3835
定价: 42.00 元
ISBN: 7900101373



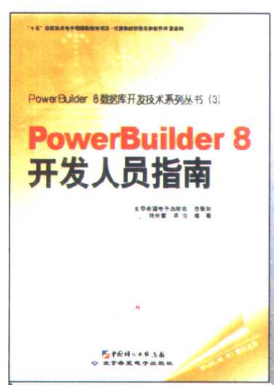
CX-3769
定价: 33.00 元
ISBN: 7900101063



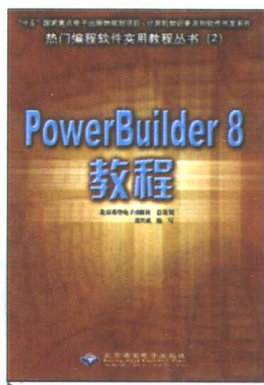
CX-3702
定价: 78.00 元
ISBN: 7900088830



CX-3703
定价: 42.00 元
ISBN: 7900088849



CX-3839
定价: 66.00 元
ISBN: 7900101411

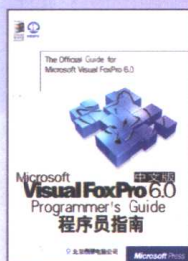


CX-3852
定价: 33.00 元
ISBN: 7900101810

社址: 北京海淀知春路 63 号卫星大厦三层 通信地址: 北京中关村 083 信箱 (100080)
电话: (010) 82675588 (总机) 传真: (010) 62520573

此为试读, 需要完整 PDF 请访问: www.eitongbook.com

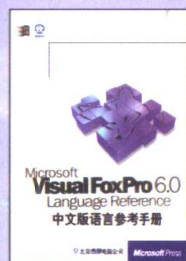
读权威书、走程序员之路



CX-2462

定价: 98.00 元

ISBN: 798002124X



CX-2467

定价: 138.00 元

ISBN: 7980021215



CX-2871

定价: 40.00 元

ISBN: 7980026306



CX-2463

定价: 98.00 元

ISBN: 7980021223



CX-2464

定价: 120.00 元

ISBN: 7980021207



CX-2477

定价: 50.00 元

ISBN: 7980023137



CX-2476

定价: 68.00 元

ISBN: 7980023129



CX-2486

定价: 90.00 元

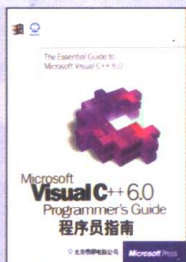
ISBN: 7980021274



CX-2597

定价: 110.00 元

ISBN: 7980023471



CX-2468

定价: 68.00 元

ISBN: 7980021231



CX-2469

定价: 70.00 元

ISBN: 7980023064



CX-2598

定价: 140.00 元

ISBN: 7980021401



CX-2599

定价: 160.00 元

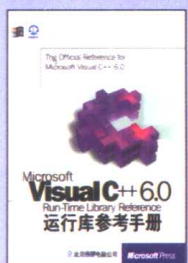
ISBN: 7980023196



CX-2723

定价: 35.00 元

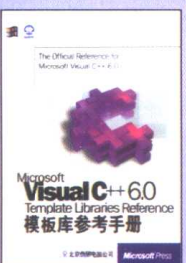
ISBN: 7980023161



CX-2480

定价: 90.00 元

ISBN: 7980023099



CX-2478

定价: 110.00 元

ISBN: 7980023102



CX-2852

定价: 66.00 元

ISBN: 7980026594



CX-2472

定价: 60.00 元

ISBN: 7980021266



CX-2473

定价: 98.00 元

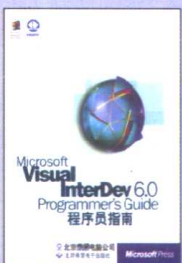
ISBN: 7980023072



CX-2479

定价: 120.00 元

ISBN: 7980023111



CX-2470

定价: 50.00 元

ISBN: 7980023080



CX-2790

定价: 68.00 元

ISBN: 7980021460



CX-2652

定价: 95.00 元

ISBN: 7980023153



CX-2530

定价: 30.00 元

ISBN: 7801440897



CX-2800

定价: 98.00 元

ISBN: 7900024514



CX-2801

定价: 169.00 元

ISBN: 7900024603



CX-2802

定价: 95.00 元

ISBN: 7900024611



CX-2803

定价: 135.00 元

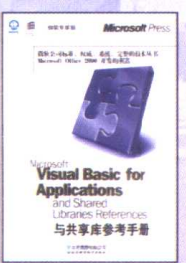
ISBN: 790002462X



CX-2804

定价: 100.00 元

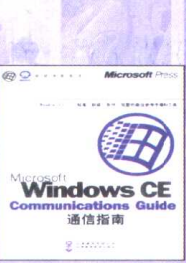
ISBN: 7900024638



CX-2805

定价: 125.00 元

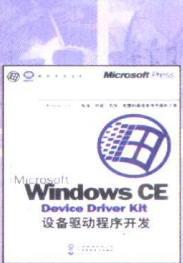
ISBN: 7900024557



CX-2808

定价: 35.00 元

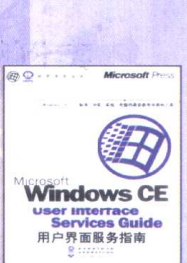
ISBN: 7900024662



CX-2811

定价: 25.00 元

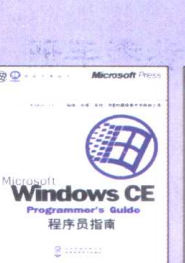
ISBN: 7900024565



CX-2809

定价: 30.00 元

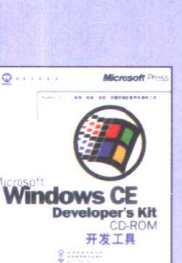
ISBN: 7900024530



CX-2810

定价: 35.00 元

ISBN: 7900024654



CX-92832

定价: 20.00 元

ISBN: 7900024980



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

通信地址: 北京中关村 083 信箱北京希望电子出版社 (邮编 100080)
联系电话: 010-82675588(总机) 传真: 010-62520573
E-mail: qrh@bhp.com.cn, 更多的信息请访问 www.bhp.com.cn

目 录

第 1 章 获取网络配置信息	1
1.1 技术提要.....	1
1.2 实例 1——获取本地主机名.....	6
1.3 实例 2——获取主机所注册的域.....	9
1.4 实例 3——获取主机名及 IP 地址	11
1.5 实例 4——根据主机名获取 IP 地址	16
第 2 章 获取网卡物理地址	20
2.1 技术提要.....	20
2.2 实例——获取网卡的物理地址.....	26
第 3 章 获取工作组及用户信息	31
3.1 技术提要.....	31
3.2 实例 1——获取网络用户全名及说明信息.....	36
3.3 实例 2——枚举用户列表及用户信息.....	40
3.4 实例 3——枚举本地组.....	46
3.5 实例 4——枚举本地组及其说明.....	49
第 4 章 获取工作站配置与用户信息	53
4.1 技术提要.....	53
4.2 实例 1——获取工作站配置信息.....	55
4.3 实例 2——获取工作站用户登录信息.....	59
4.4 实例 3——获取工作站配置及用户信息.....	63
4.5 实例 4——获取工作站的配置和当前用户信息.....	68
第 5 章 获取网络机器的服务信息	74
5.1 技术提要.....	74
5.2 实例 1——枚举本机的服务.....	82
5.3 实例 2——枚举本机服务的高级应用.....	89
5.4 实例 3——枚举本地或远程机的服务.....	97
第 6 章 浏览器编程	109
6.1 技术提要.....	109
6.2 实例 1——调用浏览对话框.....	115

6.3	实例 2——浏览对话框特性的高级应用	117
6.4	实例 3——实现文件夹的预先选择	131
6.5	实例 4——查找常用的 Shell 文件夹	140
6.6	实例 5——查找网络机器或共享资源	147
第 7 章	获取网络连接信息	151
7.1	技术提要	151
7.2	实例 1——确定网络连接是否可用	153
7.3	实例 2——确定网络的连接状态	156
7.4	实例 3——确定网络目的地是否可达	160
第 8 章	获取网络资源信息	164
8.1	技术提要	164
8.2	实例 1——枚举本地网络资源	175
8.3	实例 2——枚举远程机器上的共享资源	180
8.4	实例 3——连接网络资源	187
第 9 章	TCP/IP 协议编程	193
9.1	技术提要	193
9.2	实例 1——获取 TCP 连接表	205
9.3	实例 2——获取机器的 TCP 统计数据	209
9.4	实例 3——确定网卡的本地 IP 地址	212
9.5	实例 4——获取网卡的所有本地 IP 地址	216
9.6	实例 5——获取 IP 地址列表	219
9.7	实例 6——如何通过 IP 地址获取主机名	222
9.8	实例 7——获取网络接口表	224
9.9	实例 8——基于 Winsock 的 TCP_IP 会话程序	231
第 10 章	获取服务器信息	242
10.1	技术提要	242
10.2	实例 1——枚举域中的服务器及机器	247
10.3	实例 2——枚举域中的服务器及版本信息	251
10.4	实例 3——枚举服务器会话	257
10.5	实例 4——用函数 NetMessageBufferSend 发送消息	262
第 11 章	Ping 与路由追踪	268
11.1	技术提要	268
11.2	实例 1——通过主机名 Ping 一个 IP 地址	270
11.3	实例 2——用 VB 实现路由追踪	278
11.4	实例 3——使用路由追踪解析主机名	288

第 12 章 DHCP 编程.....	298
12.1 技术提要.....	298
12.2 实例 1——确定系统是否启用了 DHCP	300
12.3 实例 2——获取 DHCP 服务器的 IP 地址.....	303
12.4 实例 3——获取所有 DHCP 服务器的 IP 地址.....	306
12.5 实例 4——发布与更新 DHCP 的 IP 地址.....	310
第 13 章 URL 编程.....	320
13.1 技术提要.....	320
13.2 实例 1——获取 URL 中的各个组成要素	324
13.3 实例 2——创建符合 URL 规范的字符串	327
13.4 实例 3——编码 / 解码 URL 中的空格字符	330
13.5 实例 4——由 DOS 路径创建本地的 URL.....	333
附录 A VB 中调用 Windows API	336

第 1 章 获取网络配置信息

程序员要想轻松地在网络中遨游,首先必须对本地主机的网络配置信息有清晰的了解,如主机名、IP 地址、所注册的域等。在网络编程中,一般都需要获取本地主机的主机名和 IP 地址,起码应该将这些信息显示给用户,让其感到程序界面的友好性。

本章将通过几个例子,演示如何获取本地计算机的主机名、IP 地址及其所注册的域。最后利用 WinSock API 函数编程演示如何根据主机的 IP 地址来获取主机名。

1.1 技术提要

首先介绍 API 函数——GetNetworkParams,通过本函数,可以非常容易地获取本地主机名及其所注册的域等信息。获取本地主机名及 IP 地址还可通过 WinSock API 编程实现,应用程序在使用 WinSock API 之前,必须先成功调用函数 WSASStartup;而应用程序结束时,又必须调用 WSACleanup 函数进行清理工作,以释放其占用的资源。

(1) 函数 GetNetworkParams

该函数返回本地计算机的网络参数。函数原型如下:

```
DWORD GetNetworkParams(  
    PFIXED_INFO pFixedInfo,    //接收数据的缓冲区指针  
    PULONG pOutBufLen          //缓冲区大小  
);
```

函数参数说明

pFixedInfo: 一个指向 FIXED_INFO 结构的指针,用于接收返回的本地计算机的网络参数。

pOutBufLen: 一个指向 ULONG (无符号长整形) 变量的指针,该变量给出了结构 FIXED_INFO 的大小。如果缓冲区太小,不足以存放返回的信息,则本变量返回实际需要的缓冲区大小,此时,函数返回 ERROR_BUFFER_OVERFLOW 错误代码。

如果函数调用成功,则返回 ERROR_SUCCESS;否则,返回下列错误代码之一:

- ERROR_BUFFER_OVERFLOW

参数 pOutBufLen 指定的缓冲区长度太小,不足以存放适配器信息。函数返回时,参数 pOutBufLen 将给出实际需要的缓冲区的大小。

- ERROR_INVALID_PARAMETER

出现本错误的原因有三个:

- 参数 pOutBufLen 为空值;
- 调用本函数的进程不能对由参数 pOutBufLen 指定的缓冲区进行读/写操作;
- 调用函数的进程不能对由 pFixedInfo 指定的缓冲区进行写操作。

- ERROR_NO_DATA

本地计算机中没有适配器信息。

- ERROR_NOT_SUPPORTED

本地计算机的操作系统不支持 GetNetworkParams 函数调用。

对于发生的错误, 可使用函数 FormatMessage 来获取错误的描述信息。

(2) 结构 FIXED_INFO

FIXED_INFO 结构包含了计算机中所有接口的信息数据。结构定义如下:

```
typedef struct {
    char HostName[MAX_HOSTNAME_LEN + 4] ;
    char DomainName[MAX_DOMAIN_NAME_LEN + 4];
    PIP_ADDR_STRING CurrentDnsServer;
    IP_ADDR_STRING DnsServerList;
    UINT NodeType;
    char ScopeId[MAX_SCOPE_ID_LEN + 4];
    UINT EnableRouting;
    UINT EnableProxy;
    UINT EnableDns;
} FIXED_INFO, *PFIXED_INFO;
```

结构成员说明

HostName[MAX_HOSTNAME_LEN + 4]: 本地计算机的主机名。

DomainName[MAX_DOMAIN_NAME_LEN + 4]: 本地计算机注册的域名。

CurrentDnsServer: 当前的 DNS 服务器。

DnsServerList: IP_ADDR_STRING 结构链表, 给出了本地计算机使用的所有 DNS 服务器。

NodeType: 指明了本地计算机的结点类型。它可为下列值之一:

- BROADCAST_NODETYPE
- PEER_TO_PEER_NODETYPE
- MIXED_NODETYPE
- HYBRID_NODETYPE

ScopeId[MAX_SCOPE_ID_LEN + 4]: 指定了 DHCP 范围名。

EnableRouting: 指定本地计算机是否允许路由。

EnableProxy: 指定本地计算机是否作为 ARP 代理服务。

EnableDns: 指定本地计算机是否允许 DNS。

(3) 函数 WSASStartup

函数 WSASStartup 对 WinSock DLL 进行初始化, 协商 WinSock 的版本支持, 并分配必要的资源。如果在调用 WinSock 函数之前, 没有加载 WinSock DLL, 则会返回 SOCKET_ERROR 错误, 错误信息是 WSANOTINITIALISED。本函数的原型如下:

```
int WSASStartup(
```

```
WORD wVersionRequested,
LPWSADATA lpWSADATA
);
```

函数参数说明

wVersionRequested: 指定准备加载的 WinSock 库的版本。其高位字节指定所需要的 WinSock 库的次版本号，而低位字节则指定主版本号，然后用宏 MAKEWORD(X 是高位字节, Y 是低位字节) 获得正确值。

LpWSADATA: 本参数是指向 WSADATA 结构的指针，该结构包含了加载的库版本有关的信息，它的定义如下：

```
typedef struct WSADATA{
    WORD            wVersion;
    WORD            wHighVersion;
    char            szDescription[WSADESCRIPTION_LEN+1];
    char            szSystemStatus[WSASYS_STATUS_LEN+1];
    unsigned short  iMaxSockets;
    unsigned short  iMaxUdpDg;
    char FAR*       lpVendorInfo;
}WSADATA, *LPWSADATA;
```

该结构中各参数意义如下。

wVersion: 表示准备使用的 WinSock DLL 版本。

wHighVersion: 返回现有的 WinSock DLL 的最高版本。

szDescription 和 szSystemStatus: 由特定的 WinSock 实施方案设定，实际上这两个成员没有用。

iMaxSockets: 表示可同时打开的套接字数目。可同时打开的套接字的最大数目不是固定的，它很大程度上取决于物理内存的大小。

这个数字可以很好地反映出 Windows Sockets DLL 或网络软件的配置方式。程序员可以利用这个成员对能否使用 Windows 套接字进行初步判断。例如，在启动时，某个 Windows 服务器可以检查 iMaxSockets 成员，如果它小于 8，则应用程序可以显示一个错误信息以让用户重新配置网络软件。一个应用程序不应分配 iMaxSockets 个套接字，因为可能还有其他 Windows Sockets 应用程序会使用套接字。

iMaxUdpDg: 表示数据报的最大长度，以字节为单位。一般不要使用它，若想获得数据报的最大长度，应通过函数 WSAEnumProtocols 来查询协议。

如果不限数据长度，则设置本成员值为 0。对于大多数的伯克利套接字 (Berkeley sockets) 应用程序而言，默认的限制是 UDP 报文不超过 8192 个字节。Windows 套接字应用程序可以通过为报文分配指定大小缓存来强行限制报文长度，一个复杂的 Windows 套接字应用中的 iMaxUdpDg 的最小值为 512 字节。

lpVendorInfo: 本成员是为 WinSock 实施方案有关的指定厂商信息预留的，任何一个 Win 32 平台上都没有使用本成员。

(4) 函数 WSACleanup

函数 WSACleanup 终止对 WinSock DLL 的使用，并释放资源，以备下一次使用。函数原型如下：

```
int WSACleanup(void);
```

该函数不带任何参数，若函数调用成功则返回 0，否则返回错误。

(5) 函数 WSAGetLastError

该函数可以获得上次失败操作的错误状态。函数原型如下：

```
int PASCAL FAR WSAGetLastError(void);
```

本函数返回上次发生的网络错误。当一特定的 Windows Sockets API 函数指出已经发生了一个错误，则应调用本函数来获得相应的错误代码。

(6) 函数 gethostname

该函数可以返回本地主机的标准主机名。函数原型如下：

```
int PASCAL FAR gethostname(
    char FAR *name,
    int      namelen
);
```

函数参数说明

name: 一个指向要存放主机名的缓冲区指针。

namelen: 缓冲区的长度。

该函数把本地主机名存放在由 name 参数指定的缓冲区中。返回的主机名是一个以空字符结尾的字符串。主机名的形式取决于 Windows Sockets 的实现，也就是说，它可以是一个简单的主机名，也可以是一个域名。

如果函数调用成功，则返回 0；否则返回 SOCKET_ERROR。应用程序可调用函数 WSAGetLastError 来获取详细的错误信息。

(7) 函数 CopyMemory

函数 CopyMemory 将一个内存块从一个地址复制到另一个地址。函数原型如下：

```
VOID CopyMemory(
    PVOID Destination,    //目的缓存区地址
    CONST VOID *Source,   //源缓存区的地址
    DWORD Length          //源缓存区的字节数
);
```

函数参数说明

Destination: 指向复制内存块的目的单元的起始地址。

Source: 指向将复制的内存块的起始地址。

Length: 定义所复制的内存块的大小（以字节为单位）。

该函数没有返回值。需要注意的是，如果源内存块的地址和目标内存块的地址有重叠，操作结果将无法确定。对于这种情况，可使用函数 MoveMemory。

(8) 函数 gethostbyname

Windows 套接字函数 `gethostbyname` 根据主机名称返回相应的信息。函数原型如下：

```
struct hostent FAR * gethostbyname (  
    const char FAR * name  
);
```

函数参数说明

name: 一个以空字符结尾的表示主机名的字符串。

函数 `gethostbyname` 返回一个 `HOSTENT` 结构的地址，该结构由 Windows 套接字分配。

应用程序不能修改本结构或释放它的某个成员。此外，对于每一个线程，只分配了一个该结构的副本，因此，应用程序在调用任何其他 Windows 套接字函数前，应先保存该结构中的有用信息。

函数 `gethostbyname` 不能解析传递给它的 IP 地址字符串。也就是说，如果向函数传递一个 IP 地址字符串作为参数，则函数认为这是一个未知的主机名参数。如果要想通过机器的 IP 地址字符串获取相应的主机信息，可以先调用函数 `inet_addr`，将 IP 地址字符串转换为实际的 IP 地址，然后使用函数 `gethostbyaddr` 来得到 `HOSTENT` 结构中的信息。

函数 `gethostbyname` 可以成功解析由函数 `gethostname` 返回的主机名字符串。如果成功，则函数返回一指向 `HOSTENT` 结构的指针；否则，函数返回一个 `NULL` 指针，若调用函数 `WSAGetLastError`，还可得到相应的错误号码。错误代码如下：

- **WSANOTINITIALISED**

在调用该函数前，必须首先成功调用函数 `WSAStartup`。

- **WSAENETDOWN**

网络子系统出现故障。

- **WSAHOST_NOT_FOUND**

未找到主机。

- **WSATRY_AGAIN**

未找到主机或服务。

- **WSANO_RECOVERY**

发生了不可修复的错误。

- **WSANO_DATA**

无效的主机名。

- **WSAEINPROGRESS**

在进程中发生了一个阻塞 Windows Sockets 1.1 调用，或服务仍在处理一个回调函数。

- **WSAEFAULT**

参数 `name` 所指定的机器不是用户地址空间中的有效部分。

- **WSAEINTR**

通过函数 `WSACancelBlockingCall` 取消了阻塞 Windows Socket 1.1 调用。

(9) 结构 `HOSTENT`

该结构由 Windows 套接字分配。应用程序不能修改本结构或释放它的某个成员。此外，对于每一个线程，只分配了一个该结构的副本，因此，应用程序在调用任何其他 Windows

套接字函数前，应先保存该结构中的有用信息。结构的定义如下：

```
struct hostent {
    char FAR *      h_name;
    char FAR * FAR * h_aliases;
    short           h_addrtype;
    short           h_length;
    char FAR * FAR * h_addr_list;
};
```

结构成员说明

h_name: 主机的官方名称。如果使用 DNS 或类似的解析系统，则它应是一个服务器能解析的完全合格的域名（FQDN: Fully Qualified Domain Name）。如果使用本地“hosts”文件，它表示 IP 地址后的第一个入口。

h_aliases: 可选的别名数组，该数组以 NULL 结尾。

h_addrtype: 返回的地址的类型。

h_length: 每个地址的字节长度。

h_addr_list: 一个以 NULL 结尾的主机地址列表。返回的地址按照网络字节顺序进行排序。

1.2 实例 1——获取本地主机名

调用函数 `GetNetworkParams`，返回一个 `FIXED_INFO` 结构，该结构中的 `HostName` 成员包含了本机的主机名称。这里主机名是“Tz014”，其运行界面如图 1-1 所示。

首先，在 VB 6 中新建“标准 EXE”工程，在主窗体表单上添加一个 Textbox（`TxtHostName`）控件、一个 Command（`CmdGetHostName`）控件以及一个 Label（设置其属性 `Caption = “本机的主机名称”`）控件。

然后向工程中添加一个 BAS 模块，在该模块中，将声明程序中要使用到的 API 函数及相关的变量、数据结构等。当然，也可以将这些代码添加到代码窗口中。

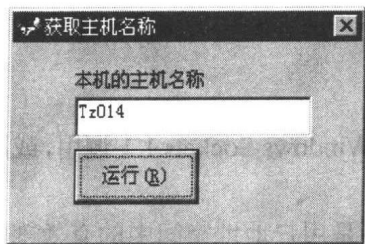


图 1-1 获取主机名

BAS 模块代码如下：

```
Option Explicit
```

```
Public Const ERROR_SUCCESS As Long = 0 '错误代码常数
```

```

'域名的最大长度
Public Const MAX_DOMAIN_NAME_LEN As Long = 128
'主机名的最大长度
Public Const MAX_HOSTNAME_LEN As Long = 128
'范围标志的最大长度
Public Const MAX_SCOPE_ID_LEN As Long = 256
Public Type IP_ADDRESS_STRING 'IP 地址
    IpAddr(0 To 15) As Byte
End Type
Public Type IP_MASK_STRING 'IP 掩码
    IpMask(0 To 15) As Byte
End Type
Public Type IP_ADDR_STRING 'IP 地址
    dwNext As Long
    IpAddress As IP_ADDRESS_STRING
    IpMask As IP_MASK_STRING
    dwContext As Long
End Type

'定义 FIXED_INFO 结构
Public Type FIXED_INFO
    HostName(0 To (MAX_HOSTNAME_LEN + 3)) As Byte '主机名
    DomainName(0 To (MAX_DOMAIN_NAME_LEN + 3)) As Byte '域名
    '当前 DNS (域名服务器) 的 IP 地址
    CurrentDnsServer As IP_ADDR_STRING
    DnsServerList As IP_ADDR_STRING 'DNS 服务列表
    NodeType As Long '节点类型
    ScopeId(0 To (MAX_SCOPE_ID_LEN + 3)) As Byte '范围标志
    EnableRouting As Long '允许使用路由器
    EnableProxy As Long '允许使用 PROXY 代理
    EnableDns As Long '允许使用 DNS
End Type

'说明 API 函数
Public Declare Function GetNetworkParams Lib "iphlpapi.dll" _
    (pFixedInfo As Any, pOutBufLen As Long) As Long

Public Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" _
    (Destination As Any, Source As Any, ByVal Length As Long)

```

完成了对 API 函数和相关数据的声明后，下面运用这些函数和数据来获取本机的主机

名称。将工作窗口切换到 Form1 表单的代码窗口，添加两个私有函数：GetHostName()（注意，它只是自定义函数，与 WinSock 的函数不同）和 TrimNULL()。函数 GetHostName() 通过调用 API 函数来取得本机的主机名称；函数 TrimNULL() 的主要功能是截取字符串中的空字符前的子字符串。代码如下：

```
Private Function GetHostName() As String
    Dim buff() As Byte
    Dim cbRequired As Long
    Dim nStructSize As Long
    Dim Info As FIXED_INFO
    '以空 pFixedInfo 参数调用 API 函数
    '这样该 API 函数会返回存放数据所需的缓存大小
    '该值被保存在参数 cbRequired 中
    Call GetNetworkParams(ByVal 0&, cbRequired)
    If cbRequired > 0 Then
        '创建所需大小的缓存
        ReDim buff(0 To cbRequired - 1) As Byte
        '再一次调用 API 函数
        If GetNetworkParams(buff(0), cbRequired) = ERROR_SUCCESS Then
            '拷贝该缓存中的数据为 FIXED_INFO 类型
            CopyMemory Info, ByVal VarPtr(buff(0)), LenB(Info)
            '返回主机名称
            GetHostName = TrimNULL(StrConv(Info.HostName, vbUnicode))
        End If
    End If
End Function

Private Function TrimNULL(item As String)
    Dim pos As Integer
    '检查字符串中是否有 chr$(0) 字符(即空字符)
    pos = InStr(item, Chr$(0))
    If pos Then
        TrimNULL = Left$(item, pos - 1)
    Else
        TrimNULL = item
    End If
End Function

最后，添加响应命令按钮的鼠标单击事件的函数，在该函数中添加如下代码：
Private Sub CmdGetHostName_Click()
    TxtHostName = GetHostName()
End Sub
```