

Visual Basic 与 Windows API 程序设计高手



黄嘉辉 黄悦珊 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



Visual Basic 与 Windows API 程序设计高手

黄嘉辉 黄悦珊 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书通过 Visual Basic 介绍如何运用 Windows API 开发高级 Windows 程序,共分 10 章。全书通过 Shell API、Messaging API 和 Messaging API 三大主题详细介绍 Windows API 在 Visual Basic 编程中的应用。

Shell API 的应用除了介绍 Windows 经过改版至 95/NT/98/2000 所新增的 Shell API 之外还介绍了 Microsoft 未公开的 Shell API 和与 Internet Explorer 有关的 API; Messaging API 的应用以介绍与 E-mail 有关的 MAPI 和支持 E-mail 的 CDO 为主; Networking API 的应用以介绍支持远程访问拨号网络服务的 RASAPI32、支持网络资源应用的 Wnet API 和以支持 HTTP、FTP 及 Gopher 等通讯协议的 Win32 Internet API 为主,还介绍了与网络传输基础 Socket 有关的 Winsock API。

本书实例丰富、详细,是一本很好的学习参考书,适合熟悉 Visual Basic 的网络开发人员学习使用。

本书繁体字版名为《Visual Basic 与 Windows API 进阶程式设计》,由文魁资讯股份有限公司出版,版权属黄嘉辉、黄悦珊所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号:图字 01-2001-3795 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Visual Basic 与 Windows API 程序设计高手

作 者: 黄嘉辉 黄悦珊

责任编辑: 石宏宇

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 25.75 字数: 644 千字

版 次: 2001 年 10 月第 1 版 2001 年 10 月第 1 次印刷

书 号: ISBN 7-900637-17-6

印 数: 0001~5000

定 价: 48.00 元(含 1 张光盘)

序

本书通过 Visual Basic 介绍如何运用 Windows API 开发高级 Windows 程序。

Visual Basic 自推出以来已经改版多次，在作者完稿之时，Visual Studio.NET 的试用版也已经出来了，在此时推出这本有关 Windows API 的书，或许有人会问是否有些晚？作者也曾经思考过这个问题，况且有关 Visual Basic 的书实在太多了，可概分为以下几类：

- 介绍基本控件的书，属于入门类。
- 与数据库有关的书，主要介绍 ADO 等。
- 介绍 Windows API 的书，但大多数都讨论相同的内容，如 GDI、Registry、Windows Message 等基本结构。
- 介绍 COM 和 COM+高级程序设计的书。
- 介绍 Internet Programming 的书，如通讯协议的应用等。

作者一直希望能写一本深入研究 Windows API 但有别于一般的介绍 Windows API 的书。但由于 Windows API 的内容实在太多了，因此本书将主要讨论 3 大主题，分别为：Shell API（第 2、3、4 章）、Messaging API（第 5、6 章）、Networking API（第 7、8、9、10 章）。

本书各章的主要内容如下：

- 第 1 章 Windows API 简介，简要介绍 Windows API 在 Visual Basic 编程中的应用。
- 第 2 章 主要介绍 Windows 经过改版至 95/NT 以及 98/2000 后所新增的 Shell API。
- 第 3 章 主要讨论 Microsoft 未公开(即 MSDN 没有记载)的 Shell API。
- 第 4 章 自从 Microsoft 将 Internet Explorer 和 Windows 系统完全整合之后，新增了不少 Windows API，本章主要介绍与 Internet Explorer 有关的 API。
- 第 5 章 介绍与 E-mail 有关的 MAPI(Messaging API)。
- 第 6 章 主要介绍与 E-mail 有关的 CDO(Collaboration Data Object)。
- 第 7 章 主要介绍支持远程访问(Remote Access)拨号网络服务的 RASAPI32。
- 第 8 章 介绍支持网络资源应用 (Windows Networking) 的 Wnet API。
- 第 9 章 介绍支持 HTTP、FTP 和 Gopher 等协议为主的 Win32 Internet API。
- 第 10 章 网络传输的基础就是 Socket,本章将介绍 Microsoft 根据 BSD(Berkeley Socket Distribution)所开发出来的 Winsock API。

本书以范例为主，通过大量的详实范例介绍如何在用 Visual Basic 编程的过程中调用 API 函数实现特殊的功能。本书的理论讲解深入浅出，使读者在学习范例的过程中更深入地了解 Windows API。

本书适合于熟悉 Visual Basic 的网络编程人员学习使用。

目 录

第 1 章 Windows API 简介 1

1.1 控件与 Windows API 4

1.2 Windows API 成员 5

第 2 章 Windows Shell API 9

2.1 Shell32 简介 10

2.2 Shell32 的基本应用 12

2.2.1 ShellExecute 函数 12

2.2.2 范例 2.1 14

2.2.3 范例 2.2 18

2.3 关于 Windows 20

2.3.1 ShellAbout 函数 20

2.3.2 范例 2.3 21

2.4 ShellExecute 的加强版本 24

2.4.1 ShellExecuteEX 函数 24

2.4.2 范例 2.4 26

2.5 任务栏信息 28

2.5.1 SHAppBarMessage 函数 30

2.5.2 范例 2.5 31

2.6 回收站 34

2.6.1 SHFileOperation 函数 34

2.6.2 范例 2.7 36

2.6.3 浏览回收站 39

2.6.4 SHQueryRecycleBin 函数 39

2.6.5 范例 2.8 40

2.6.6 清空回收站 41

2.6.7 范例 2.9 42

2.7 最近打开过的文档 43

2.7.1 SHAddToRecentDocs

函数 45

2.7.2 范例 2.10 45

2.8 获得文件目录属性 47

2.8.1 SHGetFileInfo 函数 47

2.8.2 范例 2.11 49

2.9 浏览文件夹 52

2.9.1 SHBrowseForFolder 函数 53

2.9.2 范例 2.14 54

2.9.3 SHGetPathFromIDList 函数 56

2.9.4 浏览默认文件夹 58

2.9.5 范例 2.15 59

2.9.6 范例 2.16 62

2.10 系统状态栏图标 63

2.10.1 Shell_NotifyIcon 函数 64

2.10.2 范例 2.17 65

2.10.3 范例 2.18 69

2.11 以 PWS 为例综合介绍

Windows Shell 71

2.11.1 PWS 72

2.11.2 范例 2.19 73

第 3 章 Microsoft 未公开的

Shell API 83

3.1 格式化磁盘 84

3.1.1 SHFormatDrive 函数 84

3.1.2 范例 3.1 85

3.2 新建文件夹 87

3.2.1 SHCreateDirectoryEX 函数 87

3.2.2 范例 3.2 88

3.3 检测磁盘的空间 88

3.3.1 SHGetDiskFreeSpace 函数 90

3.3.2 范例 3.3 90

3.4 运行 91

3.4.1 SHRunDialog 函数 91

3.4.2 范例 3.4 93

3.5 更改图标 94

3.5.1 SHChangelconDialog 函数.....	95	5.4.2 MAPIResolveName 函数.....	141
3.5.2 范例 3.5.....	96	5.4.3 MAPIMessage 结构变量.....	142
3.6 查找文件.....	98	5.4.4 MAPISendMail 函数.....	142
3.6.1 SHFileExists 函数.....	98	5.4.5 范例 5.5.....	143
3.6.2 范例 3.6.....	99	5.5 阅读 E-mail.....	146
3.7 关机(1).....	100	5.5.1 MAPIFindNext 函数.....	146
3.7.1 SHShutDownDialog 函数.....	100	5.5.2 MAPIReadMail 函数.....	147
3.7.2 范例 3.7.....	101	第 6 章 电子邮件——CDO for NTS.....	149
3.8 关机(2).....	102	6.1 CDO 介绍.....	150
3.8.1 SHRestartSystemMB 函数.....	102	6.2 CDO for NTS 对象.....	151
3.8.2 范例 3.8.....	103	6.2.1 Session 对象.....	152
第 4 章 浏览器 API.....	105	6.2.2 Folder 对象.....	153
4.1 浏览器介绍.....	106	6.2.3 Messages 对象.....	153
4.2 自动完成.....	108	6.2.4 Message 对象.....	154
4.2.1 SHAutoComplete 函数.....	109	6.2.5 AddressEntry 对象.....	154
4.2.2 范例 4.4.....	110	6.2.6 Attachments 对象.....	155
4.3 下载文件.....	116	6.2.7 Recipients 对象.....	155
4.3.1 DOFileDownload 函数.....	117	6.2.8 NewMail 对象.....	156
4.3.2 范例 4.5.....	117	6.2.9 范例 6.1.....	156
4.4 收藏夹.....	119	第 7 章 Win32 RAS API——远程	访问服务.....
4.4.1 添加到收藏夹.....	120	7.1 远程访问服务的基本概念.....	160
4.4.2 范例 4.6.....	121	7.1.1 远程连接.....	163
4.4.3 整理收藏夹.....	124	7.1.2 RAPI.....	163
4.4.4 范例 4.7.....	124	7.2 执行拨号网络远程访问程序.....	164
第 5 章 Messaging API——电	子邮件.....	7.2.1 拨号网络.....	165
5.1 电子邮件.....	128	7.2.2 范例 7.1.....	166
5.1.1 Spooling.....	128	7.2.3 InternetAutodial 函数.....	167
5.1.2 SMTP.....	131	7.2.4 范例 7.2.....	168
5.1.3 POP3.....	133	7.2.5 InternetAutodialHangup	函数.....
5.2 Microsoft Messaging API.....	133		169
5.3 开始和结束 MAPI 会话.....	136	7.2.6 InternetDial 函数.....	171
5.3.1 MAPIlogon 函数.....	136	7.2.7 范例 7.3.....	172
5.3.2 MAPILogoff 函数.....	137	7.2.8 InternetHangup 函数.....	174
5.3.3 范例 5.4.....	138	7.2.9 RasDial 函数.....	175
5.4 发送 E-mail.....	141	7.2.10 范例 7.4.....	176
5.4.1 MapiRecip 结构变量.....	141	7.2.11 范例 7.5.....	178

7.2.12	RasHangUp 函数.....	181	8.4.1	WnetAddConnection 函数.....	227
7.2.13	RasGetErrorString 函数.....	181	8.4.2	范例 8.5.....	228
7.3	检测拨号连接状态.....	182	8.5	建立网络资源连接(2).....	230
7.3.1	InternetGetConnected State 函数.....	183	8.5.1	WnetAddConnection2 函数.....	231
7.3.2	范例 7.6.....	183	8.5.2	范例 8.6.....	233
7.3.3	RasGetConnectStatus 函数.....	187	8.6	建立网络资源连接(3).....	236
7.3.4	范例 7.7.....	188	8.6.1	WnetAddConnection3 函数.....	236
7.3.5	范例 7.8.....	191	8.6.2	范例 8.7.....	237
7.4	获得拨号网络设置名称.....	195	8.7	断开网络资源连接(1).....	239
7.4.1	RasEnumEntries 函数.....	196	8.7.1	WnetCancelConnection 函数.....	240
7.4.2	范例 7.9.....	197	8.7.2	范例 8.8.....	240
7.4.3	范例 7.10.....	199	8.8	断开网络资源连接(2).....	241
7.5	获得拨号连接后的 IP 地址.....	203	8.8.1	WNetCancelConnection2 函数.....	241
7.5.1	RasGetProjectionInfo 函数.....	204	8.8.2	范例 8.9.....	242
7.5.2	范例 7.11.....	205	8.9	获得连接资源名称.....	243
7.6	处理拨号网络设置的用户 账号和密码.....	210	8.9.1	WnetGetConnection 函数.....	243
7.6.1	RasGetEntryDialParams 函数.....	211	8.9.2	范例 8.10.....	244
7.6.2	RasSetEntryDialParams 函数.....	212	8.10	获得网络连接的用户名称.....	245
7.6.3	范例 7.12.....	213	8.10.1	WnetGetUser 函数.....	245
			8.10.2	范例 8.11.....	245
			8.11	列举网络资源.....	247
			8.11.1	WnetOpenEnum 函数.....	247
			8.11.2	WnetEnumResource 函数.....	248
			8.11.3	WnetCloseEnum 函数.....	249
			8.11.4	范例 8.12.....	249
			8.12	获得 WNet API 的错误信息.....	253
			8.12.1	WnetGetLastError 函数.....	253
			8.12.2	范例 8.13.....	253
第 8 章 Windows Networking—— 网络资源应用217					
8.1	Windows Networking 的基本概念.....	218	第 9 章 Microsoft Win32 Internet API 应用255		
8.2	映射网络驱动器/浏览打印机 连接端口.....	220	9.1	Microsoft Win32 Internet API 的 基本概念.....	256
8.2.1	WnetConnectionDialog 函数.....	221	9.2	尝试连接至 Internet.....	258
8.2.2	范例 8.1.....	221	9.2.1	InternetAttemptConnect 函数.....	258
8.2.3	范例 8.2.....	222	9.2.2	范例 9.1.....	259
8.3	断开网络驱动器/打印机.....	223			
8.3.1	WnetDisconnectDialog 函数.....	224			
8.3.2	范例 8.3.....	224			
8.3.3	添加打印机向导.....	225			
8.4	建立网络资源连接(1).....	227			

9.3 检查 Internet 连接	260	9.13.1 InternetOpen 函数	293
9.3.1 InternetCheckConnection 函数	260	9.13.2 InternetCloseHandle 函数	294
9.3.2 范例 9.2	261	9.13.3 范例 9.13	294
9.4 建立拨号网络连接(1)	261	9.13.4 范例 9.14	298
9.4.1 InternetAutodial 函数	263	9.14 建立 FTP、HTTP 或 Gopher 服务	301
9.4.2 范例 9.3	264	9.14.1 InternetConnect 函数	302
9.5 建立拨号网络连接(2)	266	9.14.2 InternetCloseHandle 函数	304
9.5.1 InternetDial 函数	266	9.14.3 范例 9.15	304
9.5.2 范例 9.4	267	9.14.4 范例 9.16	308
9.6 断开拨号网络连接	269	9.15 HTTP 协议——下载网页内容	309
9.6.1 InternetAutoDialHangup 函数	269	9.15.1 InternetOpenUrl 函数	310
9.6.2 InternetHangUp 函数	269	9.15.2 InternetReadFile 函数	311
9.6.3 范例 9.5	270	9.15.3 范例 9.17	312
9.6.4 范例 9.6	271	9.15.4 InternetSetOption 函数	315
9.7 获得当前网络连接状态	272	9.15.5 范例 9.18	316
9.7.1 InternetGetConnectedState 函数	272	9.16 FTP 文件传输通讯协议的 基本介绍	318
9.7.2 范例 9.7	273	9.16.1 InternetConnect 函数	319
9.7.3 范例 9.8	274	9.16.2 范例 9.19	319
9.8 HTTP Cookies	277	9.17 浏览 FTP 服务器目录	324
9.8.1 Cookies 简介	277	9.17.1 FtpGetCurrentDirectory 函数	324
9.8.2 Win32 Internet API 在 HTTP Cookies 上的支持	279	9.17.2 范例 9.20	325
9.8.3 范例 9.9	281	9.17.3 范例 9.21	328
9.8.4 范例 9.10	282	9.17.4 范例 9.22	333
9.9 URL 介绍	284	9.17.5 小结	334
9.10 URL 正规化处理	285	9.18 FTP 下载及上传功能	334
9.10.1 什么是 URL 的 正规化处理	285	9.18.1 下载文件	335
9.10.2 范例 9.11	286	9.18.2 范例 9.23	335
9.11 合并 URL 字符串	287	9.18.3 上传文件	339
9.11.1 InternetCombineURL 函数	287	9.18.4 范例 9.24	339
9.11.2 范例 9.12	288	9.18.5 小结	342
9.11.3 其他类型的函数	290	9.19 FTP 其他功能介绍	343
9.12 Win32 Internet API 高级应用	291	9.19.1 新增 FTP 服务器端目录	343
9.13 建立 Internet 会话	292	9.19.2 范例 9.25	343
		9.19.3 删除 FTP 服务器端目录	345
		9.19.4 范例 9.26	346
		9.19.5 删除 FTP 服务端文件	348

9.19.6 范例 9.27	348	10.4.2 范例 10.4	375
9.19.7 重新命名 FTP 服务 器端文件	350	10.4.3 获得本机 IP 地址	376
9.19.8 范例 9.28	350	10.4.4 HOSTENT 的结构变量	377
9.19.9 小结	353	10.4.5 范例 10.5	378
第 10 章 Microsoft Windows		10.4.6 范例 10.6	380
Sockets API 应用	355	10.5 远程主机名称与 IP 地址的转换	381
10.1 Windows Socket 的介绍	356	10.5.1 范例 10.7	381
10.2 Winsock API 函数	359	10.5.2 范例 10.8	383
10.2.1 Winsock API 各版本之 间的兼容性	360	10.6 通讯协议信息	384
10.2.2 BSD Socket API 函数	360	10.6.1 getprotobyname 函数	384
10.2.3 Winsock 扩展函数	361	10.6.2 getprotobyname 函数	385
10.2.4 转换函数	363	10.6.3 PHOSTENT 的结构变量	386
10.2.5 数据库函数	363	10.6.4 范例 10.9	386
10.3 Winsock 应用程序的起始与结束	364	10.7 通讯协议服务信息	389
10.3.1 范例 10.1	366	10.7.1 getservbyname 函数	389
10.3.2 范例 10.2	369	10.7.2 getservbyport 函数	390
10.3.3 范例 10.3	370	10.8 处理 Winsock 错误	391
10.4 本地主机名称与 IP 地址	371	10.9 Winsock AP 高级应用介绍	392
10.4.1 获得本地主机名称	374	10.9.1 BSD 和 Winsock API	393
		10.9.2 BSD API 声明	395

第 1 章

Windows API 简介

本章要点

- 控件与 Windows API
- Windows API 成员

在可视化程序开发工具尚未推出之前,程序员要想在 Microsoft Windows 的平台上开发程序,通常都是利用 SDK(Software Development Kit)去调用系统已定义好的 API 函数(Application Programming Interface),才能编写出图形化的 Windows 程序。

但相信曾经历过这段历史的朋友一定刻苦铭心!其主要原因不外乎程序必须用 C 或 C++编写(但并非所有的用户都了解 C 或 C++)。事先还需了解各个 API 函数的意义、使用方法以及各个窗口消息和回调函数的意义等,这些因素就足可使编写 Windows 程序的兴趣大大降低。而通常写程序时,程序员身边总是会抱着一本厚厚的 Windows API Bible(圣经)以便查询。

庆幸的是,在短短几年中不断地有开发商推出“可视化程序开发工具/环境”,提供了可视化的开发环境,此类环境大多具有下列特色:

- 集成开发环境(Integrated Development Environment, IDE)

除了提供程序编辑的工具之外,还集成了程序编译、除错、运行及打包等功能。

Microsoft Visual Basic 的开发环境,如图 1.1 所示。

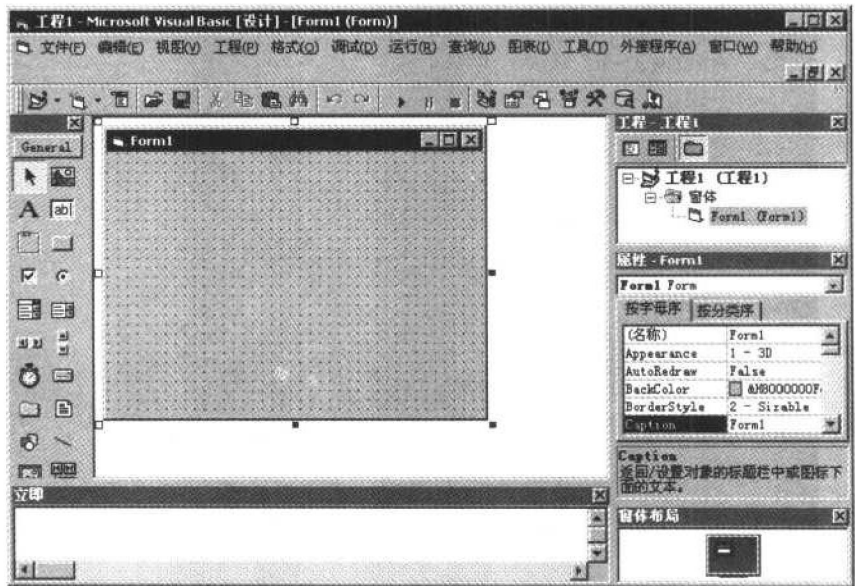


图1.1 Visual Basic的开发环境窗口

- 快速应用程序开发(Rapidly Application Development, RAD)

通过程序向导将工具箱中的控件“拖放”到窗体中,系统便在代码窗口自动产生相对应的程序代码,这样就大大减少程序开发的时间。以 Microsoft Visual Basic 为例,单击工具箱中的按钮控件并将其拖放到窗体上,在代码窗口就会自动产生相对应的程序代码,如图 1.2 所示。

- 控件化

在 API 程序设计阶段,程序员所专注的是窗口消息与回调函数的控制。但是在用控件来编写程序时,则要去了解所使用到的或是“继承”的控件所定义的属性、事件及方法。程序员只要将所要使用的控件放到窗体上并设置其相关的属性,然

后利用其方法和事件，即可编写出许多实用的应用程序，Visual Basic 中的控件如图 1.3 所示。

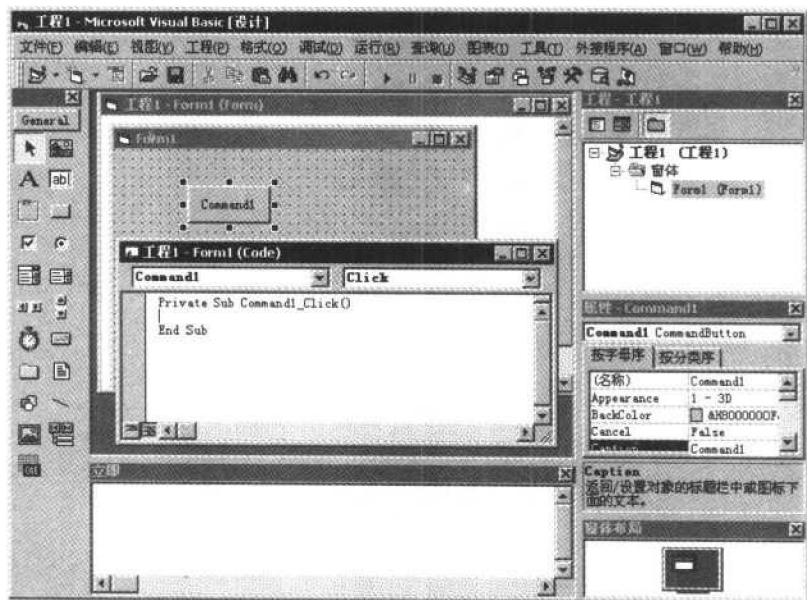


图1.2 Visual Basic的编程窗口

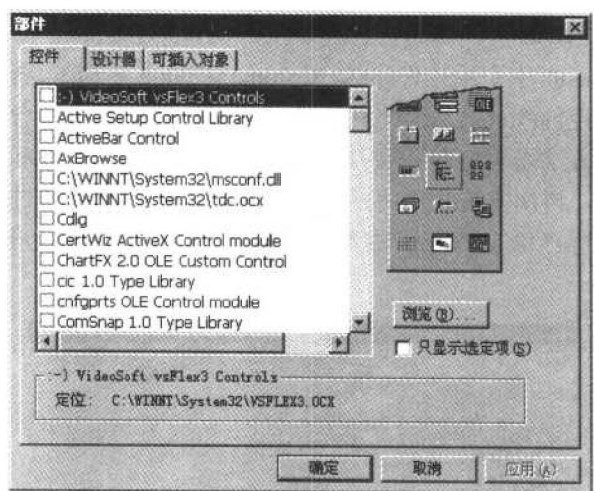


图1.3 【部件】对话框

说明：在旧版的 Visual Basic 中，其控件的文件以 “*.vbz” 表示。

在【部件】对话框中可以选取适当的控件添加到工具箱中：选定控件后，在工具箱中则出现相关的控件图标，以方便用户使用，如图 1.4 所示。

在图 1.4 中，除了 Visual Basic 默认标准的控件之外(如图片、标签、文本框、命令按

钮等), 还引用了 Visual Basic 专业版所附的控件——CommonDialog。

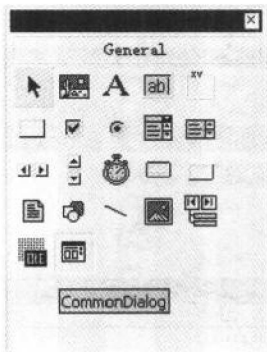


图1.4 工具箱

1.1 控件与 Windows API

既然可以使用控件, 为什么还要使用 Windows API 呢? 这是因为控件虽然方便, 但也有其缺点, 例如:

- 并不是所有的 Windows 功能, 控件都可以完成。例如本书所讨论的 Windows Networking、RAS 等, 都是控件所不支持的。
- 在 Visual Basic 中引用控件(例如 CommonDialog 控件), 虽然使用上很方便, 但由于引用到控件(*.cox), 因此其所建立的可执行文件必须含有此控件文件才能执行。解决的方法是用打包和展开向导将所有引用到的控件和执行文件打包成为一个安装程序, 再用安装程序将所引用到的控件经过安装的过程, 注册到 Windows 注册表中。所打包的 Setup 由于包含了被引用的控件而变得非常庞大, 如图 1.5 所示。

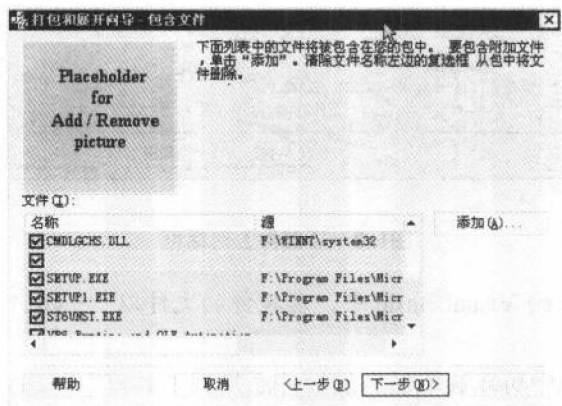


图1.5 【打包和展开向导】对话框

相反，Windows API 也有其缺点：

- 由于大部分的 API 是以 C 或 C++写成的，而 Visual Basic 与 C 在数据类型的声明上是不同的，因此需要做进一步的声明与转换，而这部分也是最容易出错的。
- 使用过 C、C++或 Delphi 的读者都知道，当引用某一 API 函数时，仅需将其已定义好的 Include 文件引用至程序中，其声明就完成了，不必自行定义。但不解的是，对于以 Basic 起家的 Microsoft，在改版多次之后，竟然在 API 函数声明上，还需要用户自行输入。即使在 Visual Basic.NET 试用版中，这部分还是一样。其实 Windows API 最难的就是函数的声明。
- 没有充分的数据供 Visual Basic 的用户查询。Microsoft 虽然提供 MSDN(Microsoft Developer Network)，但其 API 的声明几乎都以 C 或 C++的形式表示。另外像常数的声明，只有解释而没有实际的值，这就造成用 Visual Basic 声明 API 时的困难。

比较了两者之间的缺点，读者会问该使用哪个呢？答案是如果控件就能完成的，控件是不错的选择。要想实现高级功能，则必须使用 Windows API 了。

1.2 Windows API 成员

Windows API 共可分为 5 大类，分别为：

- **Function/Sub:** 函数。前者有返回值，而后者则没有。其声明方式为：

```
[Public | Private] Declare Function <函数名称>
Lib "DLL 文件名"
[Alias "函数别名"]
(argument1 As 数据类型,...) as 返回的数据类型
[Public | Private] Declare Sub <sub 名称>
Lib "DLL 文件名"
[Alias "sub 别名"]
(argument1 As 数据类型,...)
```

如：

```
Public Declare Function WNetConnectionDialog
Lib "mpr.dll"
(ByVal hwnd As Long,
ByVal dwType As Long) As Long
```

- **Type:** 结构变量，如同 C 的 Structure 类型，用以定义结构类型，其定义方式如下：

```
[Public | Private] Type <结构名称>
    argument1 As 数据类型
    argument2 As 数据类型
```

```
...  
End Type
```

如:

```
Public Type RASDIALPARAMS  
    dwSize As Long  
    szEntryName(20) As Byte  
    szPhoneNumber(128) As Byte  
    szCallbackNumber(48) As Byte  
    szUserName(256) As Byte  
    szPassword(256) As Byte  
    szDomain(15) As Byte  
End Type
```

- **常量:** Windows API 常需要指定特定的常量, 通常如指定 **Flags**, 根据不同的 **Flags** 组合, 其执行的结果也会有所不同, 其声明方式如下:

```
[Public | Private] Const <常量的名称>  
    [As 数据类型] = 常量值
```

如:

```
Public Const VER_PLATFORM_WIN32s = 0  
Public Const VER_PLATFORM_WIN32_WINDOWS = 1  
Public Const VER_PLATFORM_WIN32_NT = 2
```

- **Callback 函数:** 如同编写 Windows SDK 中的 **WinProc** 一样, **Callback** 函数用以处理 **Windows Message(WM_)**, 其声明方式通常为:

```
[Public | Private] <CallBack 函数名称>  
    (ByVal hWnd As Long,  
    ByVal uMsg As Long,  
    ByVal wParam As Long,  
    ByVal lParam As Long) As Long  
    [As 数据类型] = 常量值
```

如:

```
Public Function WinProc(ByVal hWnd As Long, _  
    ByVal uMsg As Long, _  
    ByVal wParam As Long, _  
    ByVal lParam As Long) As Long  
    Select Case lParam  
        Case WM_LBUTTONDOWNCLK  
            ...  
        Case WM_RBUTTONDOWN  
            ...  
    End Select
```

...
End Function

- **Windows Message:** 用以定义所有 Windows 执行的 Message，如单击鼠标右键或双击鼠标左键等。

