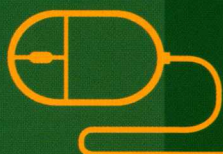


资深软件开发专家多年经验结晶，深入浅出阐释Office VBA开发涉及的工具、方法和实践
由浅入深剖析Office VBA开发过程中遇到的各个层面的问题，涉及文件和路径操作、自定义Office界面、正则表达式、字典、数据库与SQL、跨Office组件编程、工程引用与外部对象、网页自动化等核心内容

Office

VBA开发经典



中级进阶卷

刘永富 刘行◎著



清华大学出版社



Office

VBA开发经典

中级进阶卷

刘永富 刘行◎著

清华大学出版社
北京

内 容 简 介

资深软件开发专家根据自己十余年 VBA 开发经验,通过本书深入阐释 Office VBA 开发。相比于基础入门卷,本书的内容体系更加完善,知识点更高阶,以 VBA 中添加和使用外部引用为主线,详细讲述使用 VBA 操作和读写 Office 文档之外的内容,案例丰富,让读者身临其境,体会 VBA 编程的策略和魅力。

本书内容丰富、实用性强,实例典型且有代表性,可以帮助读者轻松熟悉 VBA 编程,系统学习 VBA 编程的每个层面。全书分为 14 章,内容包括文件和路径操作、文件系统自动化、压缩文件处理、XML 操作、自定义功能区、正则表达式使用方式、字典使用方法、数据库操作、Office VBA 混合编程、工程引用与外部对象、Acrobat 对象操作、邮件处理、网页自动化等。书中所有章节涉及的程序代码都给出了详细注释。

本书可作为职场办公人员、高校理工科师生、Office 专业开发人员自学用书,也可作为 Office 编程培训讲师的教学参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Office VBA 开发经典——中级进阶卷 / 刘永富, 刘行著. —北京: 清华大学出版社, 2019.
ISBN 978-7-302-52637-7

I. ① O… II. ①刘… ②刘… III. ① BASIC 语言—程序设计 IV. ① TP312.8

中国版本图书馆 CIP 数据核字(2019)第 046879 号

责任编辑:秦 健
封面设计:李召霞
责任校对:徐俊伟
责任印制:丛怀宇

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者:三河市铭诚印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:26.5 字 数:612 千字

版 次:2019 年 7 月第 1 版 印 次:2019 年 7 月第 1 次印刷

印 数:1~2000

定 价:99.00 元

产品编号:080251-01

Foreword 前言

Office VBA 编程在全世界范围已经非常普及和流行，办公人员轻轻地按下【Alt+F11】快捷键，通过录制宏就能开启 VBA 编程之旅。

VBA 的最简单应用莫过于循环处理工作簿、工作表和单元格数据。然而，Excel 中的数据只是日常办公的一小部分内容，很多数据存储在各种各样的文件中，例如记事本文件、CSV 文件、PDF 文件、数据库等，我们不得不每天双击打开一个文件，手工编辑修改，然后关闭保存。随着大数据时代、信息化时代的迅速发展，自动化、智能化办公成为大势所趋。为此，我们要对手动办公说：“No！”

作者根据自身工作经历，深切体会到文件系统自动化的必要性和重要性，用 VBA 操作 Excel 对象还不能充分发挥 VBA 的魅力。因此，本书以文件（Files）为操作对象，以外部引用（Reference）为操作手段，深刻阐述：

- 文本文件的读写。
- 文件、路径的增删。
- 注册表的读写。
- 压缩和解压缩。
- XML 与 Office 界面。
- 正则表达式处理字符串。
- 字典的特色和作用。
- ADO 访问数据库、SQL 查询语句。
- 不同 Office 组件的互相访问和通信。
- 工程引用与外部对象。
- Acrobat 软件和 PDF 文档自动化。
- 自动发送邮件。
- 网页自动化和网页数据获取。

这是进一步提高 VBA 编程技术的必经之路。

本书的组织结构

全书共分为 14 章。

第 1 章介绍文本文件的多种读写方式、路径和文件的自动化处理。

第 2 章介绍自动启动其他应用程序的方法、注册表的读写、环境变量的读写方法。

第 3 章分别介绍用 WinRAR 和 Shell32 自动处理压缩文件，以压缩文件的视角认识 Office 文档。

第 4 章介绍 XML 的语法规则，使用 XML DOM 对象读写 XML 文件，XML 文件的验证等。该章是 Office 自定义界面的理论基础。

第 5 章介绍 Office 界面五大功能区的 customUI 技术，详细讲解了用于定制 Office 界面各种控件的技术要点。

第 6 章介绍 VBA 编程中正则表达式的用法，通过丰富的实例演示和体现了正则表达式在 VBA 编程中的强大之处。

第 7 章介绍字典的构成和特点，详细讲解键值对的添加、删除、修改、检索的技巧和注意点。

第 8 章介绍使用 ADO 技术访问 Access、Excel 数据库，讲解最常用的 SQL 语句。

第 9 章介绍跨 Office 组件编程，讲解前期绑定和后期绑定的区别，以及不同 Office 组件之间的互相访问。

第 10 章介绍 VBA 工程中引用的自动添加和移除，外部对象的创建方法。

第 11 章介绍使用 VBA 调用 Acrobat 对象，对 Acrobat 软件以及 PDF 文件进行读写。

第 12 章介绍邮箱的 SMTP 配置、使用 CDO 对象自动创建和发送邮件件。

第 13 章介绍 HTML DOM 读写网页，WebBrowser 和 Internet Explorer 实现网页自动化，以及 XMLHTTP 和 WinHttp 实现 HTTP 请求。

第 14 章介绍 VBA 编程中遇到的其他常见话题：随机数、颜色和进制转换、日期时间方面的计算。

本书的特点

- 编排合理、内容丰富。
- 针对性的实例比较多，知识点讲解透彻。
- 实用性强，讲解的知识范围面向广大计算机办公人员所需。

本书的读者对象

- 职场中使用计算机的办公人员。
- 各类学校的教师、学生、科研人员。
- 从事 VBA 开发、VSTO 开发的相关人员。

本书使用环境

本书的写作环境为 Windows 7 (32 位) + Microsoft Office 2013。不过本书内容在 Office 2010 及其以上版本均兼容。

配套资源

本书配套资源包括：

- 书中涉及的所有实例文档。
- 开发资源（编程过程中用到的工具、软件）。

扫描右侧二维码访问上述资源。



读者服务

为方便广大读者学习和探讨，读者可以通过扫描右侧二维码获得更多信息。



建议学习方法

本书各章内容具有相当的难度和深度，其中第 1、2、5 ~ 9 章为必学内容，其余 7 章为选修内容。读者可根据自身所需和兴趣选择学习。

对于书中讲述的每个知识点，读者应清楚其目的和意义，实现的思路和方法，运行结果的分析等环节。

在实际学习过程中，读者可以从书中找到对应的实例文档，直接调试、运行范例中的宏，更便捷地体会程序设计的魅力。

致读者

VBA 是微软 Office 软件配套的一门编程语言，是处理文档、电子表格的首选语言。实际上，VBA 不光能处理、解决 Office 方面的问题，通过添加外部引用，还可以操作和读写 Office 以外的很多内容。外部对象库的引入使得 VBA 如虎添翼，使用恰当的外部引用解决实际问题，让编程变得更加简单、专业。

作者根据自身多年的学习和研究经验，把平时用到最多的外部引用和对象以具体实例的形式编成此书。

本书从立意到写作、交稿历时一年之久，融入作者大量精力和心血。衷心希望广大读者能够从本书汲取营养、查漏补缺，达到学以致用效果。

致谢

在本书的编写过程中，除了刘永富、刘行外，参与编写的人员还有戴海东、曹文丽、李白等。在编写过程中难免会有疏漏之处，欢迎读者通过清华大学出版社网站 www.tup.com.cn 与我们联系，帮助我们改正提高。

在本书的出版过程中，得到了清华大学出版社策划编辑秦健的大力支持和配合，在此表示衷心感谢。另外，本书所有的编审、发行人员为本书的出版和发行付出了辛勤劳动，在此一并致谢。

刘永富

Contents 目 录

第 1 章 文件和路径操作 1	
1.1 使用传统方式..... 1	
1.1.1 获取文件或路径的属性..... 2	
1.1.2 设置文件或路径的属性..... 4	
1.1.3 判断文件或路径是否存在..... 4	
1.1.4 遍历文件和子文件夹..... 5	
1.1.5 文件的复制、移动和删除..... 7	
1.1.6 文件夹的创建和删除..... 8	
1.1.7 文本文件的读写..... 8	
1.2 二进制方式读写文件..... 12	
1.2.1 字符串与字节数组的互换..... 12	
1.2.2 文本文件的写入..... 13	
1.2.3 文本文件的读出..... 13	
1.2.4 文本文件的拆分..... 15	
1.2.5 文本文件的合并..... 16	
1.2.6 二进制文件的复制..... 16	
1.3 使用文件系统对象..... 16	
1.3.1 前期绑定..... 17	
1.3.2 后期绑定..... 18	
1.3.3 FSO 对象模型..... 18	
1.3.4 遍历磁盘分区..... 18	
1.3.5 操作文件夹..... 20	
1.3.6 文件夹拒绝访问的问题..... 22	
1.3.7 操作文件..... 24	
1.3.8 遍历文件..... 25	
1.3.9 遍历子文件夹..... 26	
1.3.10 FSO 的更多操作方式..... 28	
1.3.11 判断是否存在..... 30	
1.3.12 文本文件的读写..... 31	
1.4 使用 ADODB.Stream 实现文件 读写..... 35	
1.4.1 对象的引入..... 36	
1.4.2 读取文本文件..... 36	
1.4.3 写入文本文件..... 37	
1.4.4 利用 ADODB.Stream 下载网页 附件..... 38	
1.5 本章小结..... 39	
第 2 章 文件系统自动化 40	
2.1 Shell 函数..... 40	
2.1.1 System32 中常用的可执行 文件..... 42	
2.1.2 执行 DOS 命令..... 43	
2.1.3 认识 Shell 函数的异步..... 44	
2.1.4 处理 Shell 函数中的空格..... 45	
2.1.5 自动打开控制面板..... 46	
2.1.6 打开资源管理器..... 47	
2.1.7 注册 ocx 文件和 dll 文件..... 48	
2.1.8 结束进程..... 51	
2.1.9 自动关机..... 51	

2.2 内置注册表函数	52	3.1.1 获取 WinRAR 可执行文件 路径	82
2.2.1 GetSetting	53	3.1.2 命令和开关	83
2.2.2 SaveSetting	54	3.1.3 压缩	85
2.2.3 DeleteSetting	56	3.1.4 解压缩	87
2.2.4 GetAllSettings	56	3.1.5 删除	88
2.3 使用 WshShell 操作注册表	57	3.1.6 使用通配符	88
2.3.1 读注册表项	57	3.1.7 处理压缩包的密码	90
2.3.2 写注册表项	59	3.1.8 使用 WinRAR 修改 Office 文档	91
2.3.3 删除注册表项	60	3.2 使用 Shell32 对象	94
2.3.4 创建新项	61	3.2.1 引入 Shell32 对象	94
2.4 创建快捷方式	63	3.2.2 使用 namespace 返回文件夹	95
2.4.1 创建文件的快捷方式	63	3.2.3 文件夹选择对话框	95
2.4.2 创建网址的快捷方式	65	3.2.4 遍历文件夹中的内容	96
2.5 操作环境变量	66	3.2.5 遍历 .zip 压缩包中的内容	98
2.5.1 查看和遍历环境变量	67	3.2.6 遍历 Office 文档中的内容	98
2.5.2 新建和修改环境变量	68	3.2.7 CopyHere 方法	99
2.5.3 删除环境变量	68	3.2.8 MoveHere 方法	100
2.6 自动激活指定标题文字的窗口	69	3.2.9 处理文件覆盖	102
2.7 自动关闭的对话框	70	3.2.10 处理异步问题	103
2.8 自动发送按键	71	3.2.11 修改 Office 文档功能区	103
2.8.1 按键写法	72	3.3 本章小结	105
2.8.2 多次按同一个键	74	第 4 章 操作 XML	106
2.8.3 组合按键	74	4.1 XML 构成	106
2.8.4 特殊符号的输入	75	4.1.1 元素节点	107
2.8.5 循环中使用按键	75	4.1.2 元素的属性	108
2.8.6 关于自动按键的补充说明	77	4.1.3 节点关系	108
2.9 使用 WshNetwork 对象	77	4.1.4 文本节点	108
2.9.1 返回计算机属性	77	4.1.5 注释节点	109
2.9.2 映射网络驱动器	78	4.1.6 处理指令节点	109
2.9.3 操作打印机	79	4.2 XML 语法规则	110
2.10 本章小结	80	4.2.1 标签必须正确关闭	110
第 3 章 处理压缩文件	81		
3.1 Shell 调用 WinRAR	81		

4.2.2 严格区分大小写·····	110	4.7.2 遍历元素的文本节点·····	123
4.2.3 必须有根元素·····	110	4.7.3 遍历元素的子元素节点·····	124
4.2.4 父子元素必须正确嵌套·····	110	4.7.4 遍历元素的注释节点·····	124
4.2.5 属性值必须加引号·····	111	4.8 创建和修改 XML·····	125
4.3 查看和编辑 XML·····	111	4.8.1 创建节点·····	125
4.3.1 使用记事本程序创建 XML 文件·····	111	4.8.2 插入节点·····	126
4.3.2 使用 WebBrowser 控件显示 XML·····	111	4.8.3 移除节点·····	127
4.4 使用 DOMDocument 读写 XML·····	113	4.8.4 修改和移除节点的属性·····	127
4.4.1 引入 DOMDocument 对象·····	113	4.8.5 替换节点·····	128
4.4.2 装载本地文件·····	113	4.8.6 克隆节点·····	128
4.4.3 装载网络文件·····	114	4.9 使用 Schema 验证 XML·····	129
4.4.4 装载字符串·····	114	4.9.1 在 XSD 文件中创建规则·····	129
4.4.5 保存 XML 文件·····	115	4.9.2 配置 DOMDocument 的 Schema·····	131
4.5 DOM 对象模型·····	115	4.9.3 分析验证结果·····	132
4.5.1 节点类型·····	115	4.10 XML 与 Office 文档·····	133
4.5.2 节点对象·····	116	4.10.1 添加自定义 XML 到 Word 文档·····	133
4.5.3 节点对象的属性·····	116	4.10.2 读取 Office 文档中的自定义 XML·····	134
4.6 定位节点·····	116	4.10.3 移除 Office 文档中的自定义 XML·····	135
4.6.1 使用 ChildNodes 定位所有子节点·····	117	4.10.4 工作表导入 XML·····	135
4.6.2 使用 PreviousSibling 和 NextSibling 定位前后节点·····	118	4.11 本章小结·····	137
4.6.3 使用 ParentNode 定位父节点·····	118	第 5 章 自定义功能区·····	138
4.6.4 使用 XPath 定位到任一节点·····	119	5.1 customUI 概述·····	138
4.6.5 使用 getElementsByTagName 定位到一组元素节点·····	120	5.1.1 常用功能区·····	139
4.6.6 使用 getAttributeNode 定位到属性·····	121	5.1.2 快速访问工具栏·····	139
4.7 详细了解元素节点·····	121	5.1.3 环境功能区·····	139
4.7.1 遍历元素的属性·····	122	5.1.4 右键菜单·····	140
		5.1.5 Office 菜单·····	140
		5.1.6 手动完成 customUI 设计·····	140
		5.2 使用 customUI 软件·····	143

- 5.2.1 命名空间和 Schema 验证····· 143
- 5.2.2 Custom UI Editor····· 144
- 5.2.3 Office Ribbon Editor····· 144
- 5.2.4 Visual Studio 中的 XML Editor····· 144
- 5.2.5 Ribbon XML Editor····· 145
- 5.2.6 显示加载项用户界面错误····· 148
- 5.3 自定义常用功能区····· 149
 - 5.3.1 选项卡····· 149
 - 5.3.2 组····· 151
 - 5.3.3 按钮····· 151
 - 5.3.4 小结回顾····· 152
 - 5.3.5 复选框····· 153
 - 5.3.6 组合框····· 153
 - 5.3.7 下拉框····· 154
 - 5.3.8 文本框····· 155
 - 5.3.9 标签····· 156
 - 5.3.10 分隔线····· 156
 - 5.3.11 切换按钮····· 157
 - 5.3.12 控件箱····· 157
 - 5.3.13 控件组····· 158
 - 5.3.14 图片库····· 159
 - 5.3.15 菜单····· 160
 - 5.3.16 分裂按钮····· 161
 - 5.3.17 动态菜单····· 162
 - 5.3.18 对话框····· 164
- 5.4 常用属性详解····· 165
 - 5.4.1 id-idMso····· 167
 - 5.4.2 insertBeforeMso-InsertAfterMso····· 168
 - 5.4.3 enabled-getEnabled····· 169
 - 5.4.4 visible-getVisible····· 170
 - 5.4.5 label-getLabel····· 171
 - 5.4.6 imageMso-image-getImage····· 173
 - 5.4.7 showImage-showLabel····· 178
 - 5.4.8 onAction····· 178
 - 5.4.9 onChange-getText····· 179
 - 5.4.10 onLoad····· 180
 - 5.4.11 IRibbonUI 对象····· 181
 - 5.4.12 screentip-supertip-keytip····· 184
 - 5.4.13 size····· 184
 - 5.4.14 tag····· 185
 - 5.4.15 小结回顾····· 186
 - 5.4.16 customUI 的 XML 代码编写技巧····· 188
- 5.5 使用 Commandbars 对象操作 Office 内置控件····· 189
 - 5.5.1 获取内置控件属性····· 190
 - 5.5.2 自动执行内置控件的命令····· 191
 - 5.5.3 获取内置控件的图标····· 191
- 5.6 自定义快速访问工具栏····· 191
- 5.7 自定义环境功能区····· 193
 - 5.7.1 创建自定义选项卡····· 194
 - 5.7.2 创建自定义组和控件····· 195
- 5.8 自定义右键菜单····· 197
 - 5.8.1 修改内置控件状态····· 198
 - 5.8.2 添加自定义控件····· 198
- 5.9 自定义 Office 菜单····· 200
 - 5.9.1 自定义 backstage 视图概述····· 200
 - 5.9.2 backstage 的 XML 架构····· 201
 - 5.9.3 group 风格····· 202
 - 5.9.4 taskGroup 风格····· 204
 - 5.9.5 taskFormGroup 风格····· 205
 - 5.9.6 重要属性解释····· 208
- 5.10 更改内置控件属性····· 210
- 5.11 customUI 疑难解答····· 211
- 5.12 本章小结····· 212

第 6 章 使用正则表达式	213	7.1.7 字典的比较模式	235
6.1 正则表达式入门	214	7.1.8 字典的数据类型	236
6.1.1 引用 RegExp	214	7.2 字典的应用	237
6.1.2 创建 Regexp 对象	215	7.2.1 提取单列数据中的唯一值	237
6.1.3 模式和元字符	215	7.2.2 删除重复行	238
6.1.4 是否忽略大小写	216	7.2.3 检查字符串中是否有重 复字符	239
6.1.5 是否多行模式	217	7.3 本章小结	239
6.1.6 是否全局搜索	217		
6.2 格式验证测试	217	第 8 章 操作数据库	240
6.2.1 判断是否包含特定的字符	217	8.1 Access 数据库概述	240
6.2.2 判断源文本中是否只包含 模式	218	8.1.1 数据表设计	241
6.3 替换	219	8.1.2 字段类型	243
6.4 查找	221	8.1.3 记录维护	243
6.4.1 MatchCollection 对象	221	8.2 使用 ADO 对象操作数据库	244
6.4.2 Match 对象	222	8.2.1 Connection 对象	245
6.4.3 SubMatches 对象	224	8.2.2 RecordSet 对象	247
6.5 元字符用法详解	225	8.2.3 Field 对象	248
6.5.1 字符范围	225	8.2.4 遍历记录行	249
6.5.2 多个可选	226	8.2.5 使用 Connection.Execute 方法 执行 SQL 语句	252
6.5.3 环境修饰	226	8.2.6 使用 Command.Execute 方法 执行 SQL 语句	252
6.5.4 重复多次	227	8.3 窗体中显示查询结果	253
6.5.5 贪婪和非贪婪	228	8.3.1 ListBox 控件显示结果 记录集	253
6.6 正则表达式测试器	229	8.3.2 使用 TextBox 控件显示 单条记录	254
6.7 本章小结	229	8.3.3 使用 DataGrid 控件显示结果 记录集	256
第 7 章 使用字典	230	8.4 SQL 结构化查询语言详解	259
7.1 字典对象	230	8.4.1 使用 Select 语句查询	259
7.1.1 字典的属性和方法	231	8.4.2 使用 Where 子句进行记录 筛选	262
7.1.2 键值对的添加	232		
7.1.3 键值对的修改	233		
7.1.4 键值对的移除	234		
7.1.5 指定的键是否存在	234		
7.1.6 遍历字典	234		

8.4.3	使用 Order By 进行排序	264
8.4.4	使用 Group By 进行分类 汇总	265
8.4.5	使用 Select Into 语句把查询结果 存入新表	266
8.4.6	使用 Insert Into 语句增加 记录	267
8.4.7	使用 Delete 语句删除记录	268
8.4.8	使用 Update 语句修改记录	269
8.4.9	处理 SQL 语句中的单引号	269
8.5	修改数据库结构	270
8.5.1	自动创建新数据库	271
8.5.2	自动创建新表	271
8.5.3	字段的增加删除和修改	272
8.5.4	自动删除数据表	273
8.6	访问其他类型的数据库	273
8.6.1	连接字符串的构造	273
8.6.2	查询 Excel 工作表数据	274
8.6.3	查询 CSV、TXT 文件	276
8.6.4	文本文件的快速合并	279
8.7	本章小结	280
第 9 章 Office VBA 混合编程		281
9.1	前期绑定和后期绑定	281
9.1.1	绑定前后的变化	282
9.1.2	后期绑定方式	283
9.2	创建和获取应用程序对象	284
9.2.1	使用 CreateObject	284
9.2.2	使用 New 关键字	284
9.2.3	获取正在运行的应用程序 对象	285
9.3	代码改写技巧	286
9.3.1	Word VBA 中运行 Excel VBA 代码	286

9.3.2	处理被控组件的事件过程	288
9.4	跨组件编程实例	289
9.4.1	Word VBA 调用 Excel 工作表 函数实现英汉互译	289
9.4.2	PowerPoint VBA 调用 Excel VBA 实现自动计算	291
9.4.3	Outlook VBA 基于 Excel 数据 发送邮件	293
9.4.4	Visual Basic 6.0 读写 Excel	294
9.5	本章小结	295

第 10 章 工程引用与外部对象 296

10.1	处理 VBA 工程中的引用	296
10.1.1	引用的属性	297
10.1.2	内置引用	300
10.1.3	引用的添加	300
10.1.4	引用的移除	300
10.2	外部对象和注册表	301
10.2.1	CLSID 和 ProgID	302
10.2.2	创建新对象	303
10.2.3	VBA 中使用剪贴板	303
10.3	本章小结	304

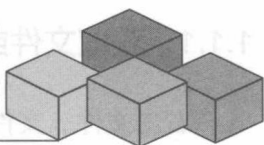
第 11 章 操作 Acrobat 对象 305

11.1	认识 Adobe Acrobat	305
11.2	Office 文档导出为 PDF 文件	307
11.2.1	Word 文档导出为 PDF 文件	308
11.2.2	Excel 工作簿导出为 PDF 文件	308
11.2.3	PowerPoint 演示文稿导出为 PDF 文件	309
11.3	Acrobat 对象模型	309
11.3.1	引用 Acrobat 对象库	309

11.3.2 Acrobat 常用对象	310	11.8.7 合并文档	329
11.3.3 Acrobat 枚举常量	311	11.8.8 替换页面	330
11.4 AcroApp 应用程序对象	312	11.9 本章小结	330
11.4.1 创建 Acrobat 对象	312	第 12 章 自动发送邮件	331
11.4.2 获取已经打开的 Acrobat 对象	312	12.1 开启 POP3/SMTP 服务	331
11.4.3 获取和设置活动工具	313	12.1.1 QQ 邮箱的 SMTP 设置	332
11.4.4 自动执行 Acrobat 工具栏控件 命令	313	12.1.2 查看邮箱服务器属性	332
11.5 AcroAVDOC 文档对象	314	12.1.3 网易 163 邮箱的 SMTP 设置	334
11.5.1 遍历所有打开的 PDF 文档	314	12.1.4 日本雅虎邮箱的 SMTP 设置	334
11.5.2 AcroAVDOC 对象的属性和 方法	315	12.2 VBA 中使用 CDO	335
11.5.3 清除选择和显示选择	316	12.2.1 配置发信账户	335
11.5.4 在 PDF 文件中查找内容	317	12.2.2 创建邮件	336
11.5.5 获取和设置 PDF 标题 文字	317	12.2.3 错误处理	338
11.5.6 获取和设置阅览模式	318	12.2.4 窗体版的邮件客户端	338
11.5.7 获取和设置 PDF 文档窗口 位置	318	12.3 其他语言调用 CDO	339
11.5.8 打印或另存 PDF 文档	318	12.3.1 VB.Net 调用 CDO	339
11.6 AcroAVPageView 对象	319	12.3.2 C# 调用 CDO	340
11.7 AcroPDPPage 对象	320	12.4 本章小结	341
11.7.1 获取和更改 PDF 页面旋转 角度	320	第 13 章 网页自动化	342
11.7.2 删除注释	321	13.1 网页自动化概述	342
11.7.3 提取页面文字	322	13.1.1 网页自动化包含的内容	342
11.8 AcroPDDoc 对象	323	13.1.2 网页自动化开发所需知识和 技能	343
11.8.1 获取和修改 PDF 文件属性	323	13.1.3 VBA 开发网页自动化的 优势	343
11.8.2 裁剪页面	324	13.1.4 本章主要内容	344
11.8.3 删除页面	327	13.2 HTML 基础	344
11.8.4 移动页面	327	13.2.1 标题	345
11.8.5 插入页面	328	13.2.2 注释	345
11.8.6 拆分文档	329	13.2.3 表格	345

13.2.4	图像	346	13.5.6	获取和操作文件资源管理器窗口	385
13.2.5	超链接	346	13.6	XMLHTTP	387
13.2.6	列表	347	13.6.1	使用 XMLHTTP 的基本流程	387
13.2.7	表单控件	347	13.6.2	判断是否联网	388
13.3	HTML DOM 对象模型	348	13.6.3	GET 和 POST 请求	388
13.3.1	使用 HTML DOM 创建网页	348	13.6.4	正确获取网页源代码	391
13.3.2	使用 HTML DOM 解析网页内容	352	13.6.5	网页中文件的下载	393
13.3.3	获取和定位网页元素	354	13.6.6	使用 API 函数下载文件	395
13.3.4	innerHTML、outerHTML、innerText、outerText 的区别	356	13.7	WinHttp	396
13.3.5	使用 InsertAdjacent 系列方法插入元素	356	13.7.1	POST 请求和响应	396
13.4	Internet Explorer 浏览器对象	358	13.7.2	抓包分析	397
13.4.1	使用浏览器的开发工具分析网页元素	360	13.7.3	构建代码	399
13.4.2	处理超链接弹出的新窗口	364	13.7.4	继续访问网站其他网页	400
13.4.3	中文字符的编码和解码	367	13.8	本章小结	401
13.4.4	使用浏览器对象的事件	368	第 14 章 其他常见话题		402
13.4.5	处理网页中的表格数据	369	14.1	随机数	402
13.4.6	自动读写表单	371	14.2	进制	402
13.5	WebBrowser 控件	374	14.3	颜色	403
13.5.1	处理 iframe	375	14.4	Excel 的文件格式	404
13.5.2	自动查看邮箱信息	378	14.5	日期和时间运算	405
13.5.3	延时等待处理	381	14.5.1	分量的提取	405
13.5.4	确保元素的获取	382	14.5.2	日期和时间的生成	405
13.5.5	获取和操作已经打开的浏览器网页	383	14.5.3	日期时间的格式化	406
			14.5.4	计算两个日期的差	407
			14.5.5	日期与数字的加减	407
			14.5.6	常见日期信息获取	408
			14.6	本章小结	409

第 1 章 文件和路径操作



我们每天使用计算机处理最多的就是文件和路径这两类对象，从 Visual Basic 的第一版至今，VB/VBA 中有关文件的处理都是通过使用 Open、Write 以及其他一些相关的语句和函数来实现的。

随着软件技术的不断发展，加上面向对象编程概念的日臻成熟，这些文件操作语句已经不能适应软件不断增加的复杂程度的需要。因此，从 VB6.0 开始，微软提出了一个全新的文件系统对象 FileSystemObject（简称 FSO）。

本章主要介绍通过如下三种方式来处理文件和路径。

- 使用传统方式。
- 使用文件系统对象 FileSystemObject (FSO) 对象。
- 使用 Shell 语句调用 DOS 命令。

本章用到的外部引用和重要对象如下。

- Microsoft Scripting Runtime
- Scripting.FileSystemObject
- Scripting.TextStream
- Microsoft ActiveX Data Objects 2.8 Library
- ADODB.Stream
- Microsoft XML, v6.0
- MSXML2.XMLHTTP60

1.1 使用传统方式

使用传统方式可以访问文件和路径，对文本文件和二进制文件进行读写。最常用的函数和命令如下。

- Dir：用于列举路径下的文件和子文件夹名称。

- ❑ GetAttr 和 SetAttr: 获取和设置属性。
- ❑ FileCopy、Name、MkDir 等: 对文件和路径复制、移动等。
- ❑ Open...Write...Close: 对文本文件、二进制文件进行打开、读写、关闭。

1.1.1 获取文件或路径的属性

右击文件、文件夹，在弹出菜单中选择属性命令，打开属性窗口后，可以设置只读属性和隐藏属性等。

GetAttr 函数用来获取和判断文件或路径的属性，该函数的参数是一个路径字符串，返回值是由多个 2 的整数幂的组合相加的总和，如表 1-1 所示。

表 1-1 文件、路径的属性常量

常 数	值	描 述
vbNormal	0	常规
vbReadOnly	1	只读
vbHidden	2	隐藏
vbSystem	4	系统文件
vbDirectory	16	目录或文件夹
vbArchive	32	上次备份以后，文件已经改变
vbAlias	64	指定的文件名是别名

这里假定磁盘下的 TE.txt 文本文件已设置为“只读”并且“隐藏”，如图 1-1 所示。



图 1-1 查看文件属性

此时, `GetAttr("C:\temp\abcd\TE.txt")` 会返回一个整数 35。其实, $35=32(\text{vbArchive})+2(\text{vbHidden})+1(\text{vbReadOnly})$ 。

因此, 把 `GetAttr` 函数的计算结果拆分为多个枚举常量值之和, 就可以得知该文件的属性。下面的过程用来把任何一个正整数拆分为多个 2 的乘方。

```
Sub SplitAttributes()  
    n = 13  
    Do Until n = 0  
        i = 2 ^ Int(Log(n) / Log(2))  
        n = n - i  
        Debug.Print i  
    Loop  
End Sub
```

运行上述过程, 可以看到 13 被拆分为 $8+4+1$ 。根据这个思路, 可以设计一个用来判断文件是否被设置为只读的自定义函数。

```
Function IsReadOnly(Path As String) As Boolean  
    n = GetAttr(Path)  
    Do Until n = 0  
        i = 2 ^ Int(Log(n) / Log(2))  
        n = n - i  
        If i = VBA.VbFileAttribute.vbReadOnly Then  
            IsReadOnly = True  
            Exit Function  
        End If  
    Loop  
End Function
```

这个函数的原理就是把 `GetAttr` 的结果拆分为多个数字, 拆分的过程中, 看看是否有一个拆分恰好等于枚举常量 `vbReadOnly`, 如果有就提前退出函数, 返回 `True`。

运行 `Debug.Print IsReadOnly("C:\temp\abcd\TE.txt")`, 在立即窗口返回结果 `True`, 表明这是一个只读文件。

同理, 把上述函数中的 `ReadOnly` 这个单词替换为 `Hidden`, 就形成了可以判断文件或路径是否设置了隐藏属性。

```
Function IsHidden(Path As String) As Boolean  
    n = GetAttr(Path)  
    Do Until n = 0  
        i = 2 ^ Int(Log(n) / Log(2))  
        n = n - i  
        If i = VBA.VbFileAttribute.vbHidden Then  
            IsHidden = True  
            Exit Function  
        End If  
    Loop  
End Function
```

这里假定 C: 盘下的 `Build` 文件夹被设置了隐藏属性, 那么 `Debug.Print IsHidden("C:\Build\")` 返回结果 `True`。