

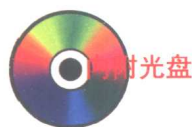


Visual C++

实践与提高

ActiveX 篇

宇鹏 王晓峰 李云飞 编著



 中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

TP312
543D

Visual C++ 实践与提高

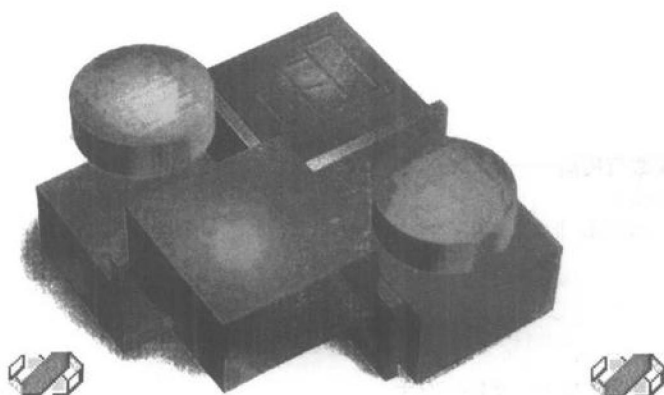
——ActiveX 篇

宇 鹏

王晓峰 编著

李云飞

 **Visual C++**



北方工业大学图书馆



00485822

中国铁道出版社

2001 • 北京

JS861 / 14

(京)新登字 063 号

内 容 简 介

本书的主要内容分为三个部分:第一部分介绍了 ActiveX 技术的一些基本知识和开发 ActiveX 所用到的一些开发工具,并对 ActiveX 的基础——COM 技术做了简要地介绍。第二部分详细讲解了如何使用 VC++ 中的微软基础类库(MFC)开发 ActiveX 的各种组件。第三部分介绍了 VC++所提供的最新 ActiveX 开发工具——ActiveX 模板库(ATL),以及如何使用 ATL 开发 ActiveX 控件和 ActiveX 自动化服务器,这部分是使用开发 ActiveX 技术的一个提高。这些都是资深 ActiveX 程序员长期积累的开发经验,它们能使读者茅塞顿开,迅速提高编程水平。

本书条理清楚、语言简练、可读性强,适用于初学者和高级程序设计人员。

图书在版编目(CIP)数据

Visual C++ 实践与提高. ActiveX /宇鹏,王晓峰,李云飞编著. —北京:中国铁道出版社,2000.12

ISBN 7-113-04006-3

I. V… II. ①宇…②王…③李… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 81007 号

书 名: Visual C++ 实践与提高——ActiveX 篇

作 者: 宇鹏 王晓峰 李云飞

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟

特邀编辑: 袁秀珍

封面设计: 冯龙彬

印 刷: 北京兴顺印刷厂

开 本: 787×1096 1/16 印张: 22.25 字数: 529 千

版 本: 2001 年 1 月第 1 版 2001 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-04006-3/TP·494

定 价: 45.00 元

版权所有 盗印必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

本书从 ActiveX 技术的基本知识开始,由浅入深地介绍了 ActiveX 技术的各个方面,并详细讲解了如何使用 VC++6.0 开发 ActiveX 组件。

本书分为三个部分,在第一部分中介绍了 ActiveX 技术的一些基本知识和开发 ActiveX 所用到的一些开发工具,并对 ActiveX 的基础——COM 技术做了简要地介绍,通过这部分的学习,使读者可以对 ActiveX 技术有一个大概的认识,为后面的实际开发打下一个良好的基础。在第二部分中,本书用了大量的篇幅,详细介绍了如何使用 VC++ 中的微软基础类库 (MFC) 开发 ActiveX 的各种组件,读者通过对这部分的学习,可以基本学会如何使用 MFC 开发 ActiveX 服务器/容器、ActiveX 自动化服务器/自动化控制器、ActiveX 控件、ActiveX 文件服务器/文件容器等等。在本书的第三部分中,介绍了 VC++ 所提供的最新 ActiveX 开发工具——ActiveX 模板库 (ATL),以及如何使用 ATL 开发 ActiveX 控件和 ActiveX 自动化服务器,这部分是使用 VC++ 开发 ActiveX 技术的一个提高。

本书的内容力求面向实际开发,着重介绍开发的方法和技巧,对原理性的内容只做一般性的阐述。在介绍开发方法时,都是结合典型的开发实例,以向导的方式循序渐进地介绍每一步的详细实现过程,使读者能够按照书中的介绍,一步步地熟悉开发的过程和方法。在介绍的过程中,还穿插介绍了一些实用的技巧以及开发过程中需要注意的地方,这些都是作者和其他资深 ActiveX 程序员长期积累的开发经验,它们能使读者茅塞顿开,迅速提高编程水平。

本书的目的,不仅仅是使读者能够完成实现书中所介绍的例子,而更重要的是使读者在读完本书之后,能对 ActiveX 技术的核心思想有一个较深刻的认识,并真正掌握实际的开发方法。

由于内容和篇幅所限,书中没有用大量的篇幅介绍 VC++ 以及 MFC 类库的使用,而是在介绍实例的开发过程中对所用到的背景知识进行简单地介绍。但是如果读者能够熟练掌握 VC++ 和 MFC 的开发方法,将大大有利于学习本书中的内容。

参加本书编写工作的有:宇鹏、于风华、唐增军、王晓峰、李云飞、葛鹏、田雪、王丹容、张翔宇、陆永年、楚天磊、孙斌、苑雪聪、孟凡贵、赵先同、吴明玉、李宏年、王天雨、赵森、钱鑫、赵伦负责本书中例子的编写,刘晖、张子介负责资料的收集整理工作。

编 者

2000 年 10 月

目 录

第 1 章 ActiveX 概述	1
1.1 ActiveX 的发展历史	1
1.2 ActiveX、OLE 和 Internet	2
1.3 ActiveX 组件的类型	3
1.3.1 ActiveX 服务器	3
1.3.1.1 什么是 ActiveX 服务器	3
1.3.1.2 ActiveX 服务器的分类	3
1.3.2 ActiveX 容器	4
1.3.2.1 什么是 ActiveX 容器	4
1.3.2.2 ActiveX 容器/服务器	4
1.3.3 ActiveX 自动化服务器	4
1.3.3.1 什么是 ActiveX 自动化	4
1.3.3.2 ActiveX 自动化服务器的分类	5
1.3.4 ActiveX 自动化控制器	6
1.3.5 ActiveX 控件	6
1.3.5.1 什么是 ActiveX 控件	6
1.3.5.2 ActiveX 控件发展的历史	6
1.3.6 COM 组件	7
1.4 选择和实现 ActiveX 组件的策略	7
1.4.1 ActiveX 自动化服务器和控制器	7
1.4.2 ActiveX 控件	8
1.4.3 COM 对象	8
1.5 小结	9
第 2 章 开发 ActiveX 的工具	11
2.1 选择合适的开发工具	11
2.1.1 Microsoft 基础类库 (MFC)	11
2.1.2 Active 模板库 (ATL)	12
2.1.3 BaseControl 框架 (BaseCtl)	12
2.1.4 建立自己的框架	13
2.1.5 开发工具小结	13
2.2 注册表和注册表编辑器	13



2.2.1	注册表发展的历史.....	14
2.2.1.1	统一记录在 SYSTEM.INI 和 WIN.INI 中.....	14
2.2.1.2	记录在自己的 INI 文件中.....	14
2.2.1.3	记录在注册表中.....	14
2.2.2	注册表的结构.....	15
2.2.2.1	注册表的结构划分及相互关系.....	15
2.2.2.2	六大根键的作用.....	16
2.2.2.3	ActiveX 组件信息在注册表中的位置.....	17
2.2.3	注册表编辑器的一些基本操作.....	17
2.2.3.1	查找.....	17
2.2.3.2	创建键和键值.....	18
2.3	开发 ActiveX 的一些实用工具.....	19
2.3.1	MIDL 编译器.....	19
2.3.2	Mktypelib 编译器.....	19
2.3.3	GUIDGEN.....	19
2.3.4	注册服务器.....	20
2.3.5	OLE/COM 对象浏览器.....	21
2.3.6	如何将工具加到 VC++ 的开发环境中.....	22
2.4	小结.....	22
第 3 章	ActiveX 的基础——COM.....	23
3.1	COM 概述.....	23
3.1.1	什么是 COM.....	23
3.1.2	COM 技术的发展历程.....	24
3.1.3	COM 结构.....	25
3.1.3.1	对象与接口.....	25
3.1.3.2	客户 / 服务器模型.....	26
3.1.3.3	COM 库.....	27
3.1.4	COM 特性.....	27
3.1.4.1	语言无关性.....	27
3.1.4.2	进程透明特性.....	28
3.1.4.3	可重用性.....	29
3.1.5	COM 对象的基本知识.....	29
3.1.5.1	COM 对象的标识——CLSID.....	30
3.1.5.2	COM 对象与 C++ 对象的比较.....	31
3.1.6	COM 接口的特点.....	32
3.1.6.1	二进制特性.....	32
3.1.6.2	接口不变性.....	33



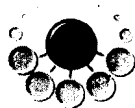
3.1.6.3 继承性 (扩展性)	33
3.1.6.4 多态性	33
3.2 COM 与 ActiveX 相关的技术	34
3.2.1 可连接对象	34
3.2.2 结构化存储	34
3.2.3 命名与绑定技术	37
3.2.4 统一数据传输	39
3.2.4.1 数据交换与传输协议的分离	40
3.2.4.2 剪贴板和拖—放操作	41
3.3 小结	42
第 4 章 使用 MFC 开发 ActiveX 服务器	43
4.1 ActiveX 服务器功能演示	43
4.1.1 单独运行程序	43
4.1.2 在注册表中登记	44
4.1.3 链接和嵌入对象	44
4.1.3.1 什么是链接和嵌入	44
4.1.3.2 嵌入对象	45
4.1.3.3 链接对象	46
4.1.4 在位编辑和开放编辑	46
4.1.4.1 在位编辑	46
4.1.4.2 开放编辑	47
4.1.5 使用剪贴板和拖—放操作	48
4.1.5.1 使用剪贴板	48
4.1.5.2 使用拖—放操作	48
4.2 实现一个最简单的 ActiveX 服务器	49
4.2.1 创建一个 ActiveX 服务器	49
4.2.2 程序结构分析	54
4.2.2.1 基本程序框架结构	54
4.2.2.2 应用程序类	54
4.2.2.3 主窗口类	55
4.2.2.4 子窗口类	55
4.2.2.5 文件类	55
4.2.2.6 视图类	56
4.2.2.7 在位框架类	56
4.2.2.8 服务器对象类	56
4.2.2.9 应用程序资源	57
4.2.3 编译运行	57
4.3 在应用程序类中初始化 ActiveX 服务器	59



4.4	在文件类中添加对服务器的支持	61
4.4.1	登记剪贴板格式	61
4.4.2	创建或编辑嵌入对象	62
4.4.3	查找链接项	63
4.4.4	设置在位编辑窗口矩形	64
4.5	实现服务器对象	65
4.5.1	绘制对象	66
4.5.1.1	在文件类中实现绘制	66
4.5.1.2	在服务器对象类中得到文件的指针	67
4.5.1.3	在服务器对象类中实现绘制	68
4.5.2	获得对象的大小	68
4.5.3	获取剪贴板对象	69
4.6	在视图类中实现在位编辑	71
4.6.1	设置绘图模式	72
4.6.2	实现绘制操作	72
4.7	在视图类中实现剪贴板操作	72
4.7.1	复制	72
4.7.2	剪切	75
4.7.3	粘贴	75
4.8	在视图类中实现拖—放操作	76
4.8.1	登记拖—放目标对象	76
4.8.2	添加成员变量记录拖—放类型	78
4.8.3	实现拖—放	79
4.8.3.1	处理鼠标左键消息	79
4.8.3.2	重载 OnDragEnter 函数	80
4.8.3.3	重载 OnDragOver 函数	80
4.8.3.4	重载 OnDragLeave 函数	81
4.8.3.5	重载 OnDrop 函数	81
4.9	小结	81
第 5 章 使用 MFC 开发 ActiveX 容器		83
5.1	概述	83
5.2	ActiveX 容器功能演示	83
5.2.1	插入对象	83
5.2.2	支持链接和嵌入对象的在位编辑	85
5.2.3	剪贴板操作	85
5.2.4	编辑和更新链接	85
5.2.5	查看和修改对象的属性	86



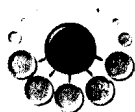
5.2.5.1	转换对象的格式	87
5.2.5.2	改变对象大小	87
5.2.5.3	更改图标	88
5.2.5.4	编辑和更新链接	89
5.3	实现一个最简单的 ActiveX 容器	89
5.3.1	创建一个 ActiveX 容器	89
5.3.2	程序结构分析	93
5.3.2.1	基本程序框架结构	93
5.3.2.2	容器对象类	94
5.3.2.3	应用程序资源	94
5.3.3	编译运行	95
5.4	实现容器对象	95
5.4.1	记录对象的位置和大小	96
5.4.1.1	添加成员变量	96
5.4.1.2	得到对象的位置和大小	96
5.4.1.3	改变对象的位置	98
5.4.2	改变容器对象的状态	99
5.4.3	存取容器对象的数据	100
5.5	实现视图类	101
5.5.1	实现矩形跟踪器	101
5.5.1.1	什么是矩形跟踪器	101
5.5.1.2	实现矩形跟踪器	101
5.5.1.3	设置矩形跟踪器的鼠标形状	103
5.5.2	实现选中对象	104
5.5.2.1	实现命中测试	104
5.5.2.2	实现选中对象	105
5.5.3	绘制对象	107
5.5.4	插入一个新的对象	108
5.5.5	设置 AxtiveX 对象的输入焦点	110
5.5.6	改变对象的大小和位置	111
5.5.7	鼠标双击编辑	112
5.5.8	删除对象	113
5.6	剪贴板操作	113
5.6.1	复制和剪切 AxtiveX 对象	113
5.6.1.1	复制对象	113
5.6.1.2	剪切对象	114
5.6.2	粘贴 AxtiveX 对象	115
5.6.2.1	实现粘贴的基本函数	115
5.6.2.2	一般粘贴	118



5.6.2.3	粘贴链接	119
5.6.2.4	选择性粘贴	119
5.7	查看对象的属性	121
5.8	实现右键快捷菜单	123
5.9	小结	123
第 6 章	用 MFC 开发 ActiveX 自动化服务器	125
6.1	创建 ActiveX 自动化服务器程序框架	125
6.1.1	创建工程	125
6.1.2	分析程序代码	127
6.1.2.1	应用程序类	127
6.1.2.2	用于 ActiveX 自动化服务器注册的三个函数	128
6.1.2.3	ActiveX 类型库	129
6.2	实现自动化服务器	130
6.2.1	添加 ActiveX 自动化接口类	130
6.2.2	添加属性和方法	134
6.2.2.1	什么是属性和方法	134
6.2.2.2	添加属性	136
6.2.2.3	添加方法	141
6.2.3	注册自动化服务器	142
6.3	小结	142
第 7 章	用 MFC 开发 ActiveX 自动化控制器	143
7.1	创建程序框架	143
7.1.1	创建工程	143
7.1.2	分析代码	146
7.2	实现 ActiveX 自动化控制器	152
7.2.1	从类型库中引入类声明	152
7.2.2	添加自动化接口类的对象	155
7.2.3	向对话框中添加控件	156
7.2.4	实现控制功能	159
7.2.5	程序演示	159
7.3	小结	160
第 8 章	使用 MFC 开发 ActiveX 控件	161
8.1	ActiveX 控件概述	161
8.1.1	ActiveX 控件的一些基本概念	161
8.1.1.1	ActiveX 控件	161



8.1.1.2	Ax tiveX 控件容器	161
8.1.1.3	事件	161
8.1.1.4	方法	162
8.1.1.5	属性	163
8.1.1.6	串行化	164
8.1.1.7	属性页	164
8.1.1.8	许可	164
8.1.2	ActiveX 控件的基本组成部分	164
8.1.2.1	基类 COleControl	164
8.1.2.2	事件激发函数	164
8.1.2.3	派遣映射表	165
8.2	在 VC++应用程序中使用 Ax tiveX 控件	165
8.2.1	创建程序框架	165
8.2.2	插入 Ax tiveX 控件	167
8.2.2.1	将 Ax tiveX 控件插入到工程中	167
8.2.2.2	插入 Ax tiveX 控件后的变化	168
8.2.3	实现功能	169
8.2.3.1	对话框布局设计	169
8.2.3.2	添加控件所对应的成员变量	171
8.2.3.3	实现具体的功能	172
8.2.4	程序演示	177
8.3	使用 MFC 开发 Ax tiveX 控件	178
8.3.1	创建基本的程序框架	178
8.3.2	代码分析	181
8.3.2.1	程序结构	181
8.3.2.2	应用程序类	182
8.3.2.3	Ax tiveX 控件类	183
8.3.2.4	Ax tiveX 控件属性页类	186
8.3.3	修改控件的图标	188
8.3.4	创建属性	189
8.3.4.1	添加公共属性	189
8.3.4.2	添加具有通知特性的属性	191
8.3.4.3	添加 Get/Set 类型的属性	193
8.3.5	为控件添加特殊效果	195
8.3.5.1	添加 FlashColor 属性	195
8.3.5.2	添加 FlashColor 函数	197
8.3.5.3	实现命中测试	198
8.3.5.4	响应鼠标消息	199
8.3.6	添加事件	200



8.3.6.1	添加自定义事件	200
8.3.6.2	激发自定义事件	203
8.3.7	绘制控件	204
8.3.8	实现属性页	205
8.3.8.1	添加公共属性页	205
8.3.8.2	实现自定义属性页	205
8.3.9	测试控件	208
8.3.9.1	插入控件	208
8.3.9.2	测试事件	209
8.3.9.3	测试自定义属性	209
8.3.9.4	测试公共属性	210
8.4	小结	211
第 9 章 使用 MFC 开发 ActiveX 文件		213
9.1	ActiveX 文件技术概述	213
9.2	使用 MFC 开发 ActiveX 文件服务器	216
9.2.1	创建 ActiveX 文件服务器项目	216
9.2.2	测试 ActiveX 文件服务器	219
9.2.3	分析 ActiveX 文件服务器程序	220
9.2.3.1	应用程序类结构	220
9.2.3.2	应用程序类	221
9.2.3.3	文件服务器类	223
9.2.3.4	在位编辑框架类	224
9.2.3.5	文件类	226
9.2.3.6	视图类	227
9.3	使用 MFC 开发 ActiveX 文件容器	227
9.3.1	创建 ActiveX 文件容器项目	227
9.3.2	测试 ActiveX 文件容器	230
9.3.3	分析 ActiveX 文件容器程序	232
9.3.3.1	应用程序类结构	232
9.3.3.2	文件对象类	233
9.3.3.3	视图类	234
9.4	小结	238
第 10 章 ActiveX 模板库概述		239
10.1	概述	239
10.1.1	什么是 ATL	239
10.1.2	ATL 的发展历史	239



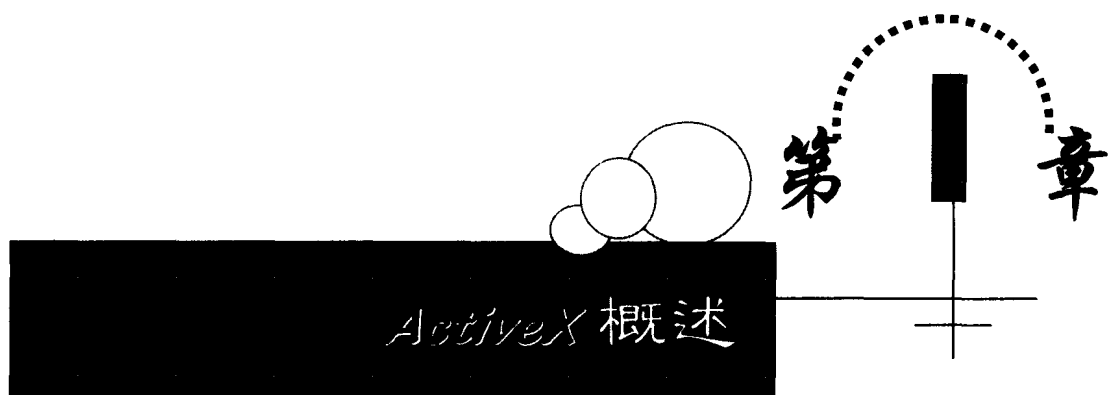
10.1.2.1	ATL 的产生	239
10.1.2.2	ATL 产生前的 COM 开发方式	240
10.1.3	ATL 的优越性	241
10.1.3.1	针对性强	241
10.1.3.2	代码简练高效	241
10.1.3.3	技术先进	241
10.1.4	ATL 的技术基础	241
10.1.4.1	COM 技术	242
10.1.4.2	C++ 模板类技术	242
10.1.4.3	多继承技术	243
10.1.5	ATL 的应用范围	243
10.2	ATL 库简明参考	244
10.2.1	ATL 简介	244
10.2.1.1	类工厂	245
10.2.1.2	COM 模块	245
10.2.1.3	连接点	245
10.2.1.4	控件支持	245
10.2.1.5	数据类型	246
10.2.1.6	接口指针	246
10.2.1.7	IUnknown 实现	246
10.2.1.8	持久性	247
10.2.1.9	属性和属性页	247
10.2.1.10	UI 支持	247
10.2.1.11	窗口支持	248
10.2.2	常用 ATL 类介绍	248
10.2.2.1	CxWindow	248
10.2.2.2	CComBSTR	249
10.2.2.3	CComClassFactory	250
10.2.2.4	CComCoClass	251
10.2.2.5	CComContainedObject	251
10.2.2.6	CComControl	252
10.2.2.7	CComDispatchDriver	254
10.2.2.8	CComModuleclass	255
10.2.2.9	CComObject	256
10.2.2.10	CComPtr	256
10.2.2.11	CComVariant	257
10.2.2.12	CContainedWindow	258
10.2.2.13	CMessageMapclass	259
10.2.2.14	CRegKey	259



10.2.2.15	CStockPropImpl	260
10.2.2.16	CWindow	261
10.2.3	常用 ATL 宏	266
10.2.3.1	COM 映射宏	266
10.2.3.2	连接点宏	266
10.2.3.3	调试和错误信息报告宏	266
10.2.3.4	消息映射宏	266
10.2.3.5	对象映射宏	267
10.2.3.6	属性映射宏	268
10.2.3.7	固定属性宏	268
10.3	小结	268
第 11 章	使用 ATL 开发 ActiveX 控件	269
11.1	开发普通 ActiveX 控件	269
11.1.1	创建 ATL 项目	269
11.1.1.1	使用应用程序向导创建 ATL 框架	269
11.1.1.2	应用程序框架分析	272
11.1.2	向 ATL 项目添加 ActiveX 控件	275
11.1.3	向 ActiveX 控件添加属性	283
11.1.3.1	添加属性	283
11.1.3.2	实现属性	285
11.1.4	修改 ActiveX 控件的绘制代码	286
11.1.4.1	计算正多边形的顶点	286
11.1.4.2	添加绘制代码	287
11.1.4.3	在改变边数时更新控件	289
11.1.5	为 ActiveX 控件添加事件	290
11.1.5.1	添加响应事件的方法	290
11.1.5.2	添加连接点接口	291
11.1.5.3	添加激发事件的代码	293
11.1.6	给 ActiveX 控件添加属性页	295
11.1.6.1	添加属性页对象	295
11.1.7	测试和发布 ActiveX 控件	299
11.1.7.1	测试事件	299
11.1.7.2	测试属性	300
11.1.7.3	测试方法	301
11.2	开发基于 Windows 标准控件的 ActiveX 控件	301
11.2.1	基于 Windows 标准控件的 ActiveX 控件	301
11.2.2	创建 ATL 项目	302



11.2.3 添加基于标准控件的 ActiveX 控件	303
11.2.3.1 添加控件	303
11.2.3.2 代码分析	304
11.2.4 实现控件	308
11.2.4.1 设置控件的风格	308
11.2.4.2 改变控件的绘制方法	309
11.2.4.3 改变控件的输入特性	310
11.2.5 测试控件	311
11.2.5.1 测试输入特性	311
11.2.5.2 测试属性	312
11.3 小结	313
第 12 章 使用 ATL 开发 ActiveX 自动化服务器	315
12.1 创建 ATL 项目	315
12.2 实现自动化服务器	316
12.2.1 添加 ATL 对象	316
12.2.1.1 使用 ATL Object Wizard 添加 ATL 对象	316
12.2.1.2 分析新添加的 ATL 对象	318
12.2.2 添加属性	320
12.2.2.1 使用 ATL 类向导添加属性	320
12.2.2.2 实现属性	322
12.2.3 添加方法	325
12.2.3.1 使用 ATL 类向导添加方法	325
12.2.3.2 实现方法	325
12.3 使用和测试自动化服务器	326
12.3.1 创建应用程序框架	326
12.3.2 添加表格对象	327
12.3.2.1 包含表格类的相关文件	327
12.3.2.2 在视图类中添加表格数据成员	331
12.3.3 生成设置对话框	332
12.3.4 通过交互生成表格	334
12.3.4.1 添加鼠标消息响应函数	334
12.3.4.2 实现交互代码	335
12.3.5 绘制表格	337
12.3.6 运行程序	338
12.4 小结	339



在最近的几年中, ActiveX 已经成为一门炙手可热的软件技术, 很多开发者和软件开发商把它作为自己的开发方向。然而, 到底什么是 ActiveX, 它对程序开发和软件技术的发展有什么意义, 却没有多少人能够真正的明白。在这一章中, 将对 ActiveX 做一个概括性地介绍。

1.1 ActiveX 的发展历史

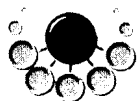
当今世界的软件技术发展可以说是日新月异, 今天的软件与以前相比, 功能越来越强大, 也越来越容易使用, 而当它们变得越来越复杂和庞大时, 工程化、维护和升级这些软件就变得越来越困难, 而且代价越来越大。增加新特性总是会引入新的、有时很难发现的错误(Bugs), 并危及到与早期版本的兼容性。此外, 很多程序是非常巨大的, 虽然提供了丰富的特性, 但是很难添加原来没有的功能或是删除已经没用的功能。后来, 有些应用程序虽然能够用插入、删除模块的方法支持添加和删除功能, 这使得他们可以在一个开发团队内部实现资源共享和协同工作, 但是要想把不同厂商所提供的应用程序组合在一起, 进行协同工作, 却是非常困难的。

操作系统也存在同样的问题。由于大多数操作系统不具有非常有效的独立性, 以一种独立、干净、灵活的方式升级、替换、重用已有的功能是非常困难的。像应用程序开发商一样, 操作系统开发商也面临这样的两难抉择: 是对原有的功能进行一些折衷的改进以保持兼容性, 还是放弃一些沉重的包袱、跟随技术的发展而进步。

20 世纪 80 年代以来, 软件开发者已经逐渐认识到面向对象的程序设计(OOP)思想是对这些问题的一个充满希望的解决方案。通过将数据和函数封装为对象, 然后通过指针和引用来访问对象中的内容, 面向对象的程序设计思想可以把一个庞大的程序逐层分解为若干功能模块, 这就使大规模的协作开发成为可能。面向对象思想进一步发展, 又相继产生了继承、多态和聚集、包容等机制, 这样就可以使新开发的对象可以充分利用已经存在的对象, 大大地提高了以往工作的重用性, 从而降低开发的成本。

即使这样, 面向对象的思想还没有充分发挥它的潜能, 因为没有了一个标准框架, 使不同软件厂商所开发的软件可以跨进程、跨平台, 甚至通过网络跨机器、跨地域地进行交互和协同工作。一旦您把软件分解成可管理、可以相互交互的组件, 就需要由一些方法、规范和标准来使它们之间可以进行通讯, 这样 OLE 就产生了。

OLE 的全名是“Object Linking and Embedding”, 意思是对象链接与嵌入。OLE 最初是



微软公司在 Windows3.1 中，为了创建和管理复合文件而推出来的，当时的版本是 OLE 1.0，随着软件技术的不断发展，在推出 Windows95 的时候同时推出了 OLE2.0，这时 OLE 的内涵已经远远不止创建和管理复合文件了，它的内容包括结构化存储、命名、统一数据传输、剪贴板、拖一放、在位编辑等等，所有这些也正是日后 ActiveX 技术的基础。OLE 的实质贡献在于，它提供了一个标准的框架用来创建、管理和访问基于对象的组件，而这些组件为其他程序提供一些功能和服务。OLE 组件不仅可以作为操作系统或应用程序的一部分存在，还可以作为一个独立的部分存在。可以这样说，OLE 技术是各种类型的功能组件之间的一种集成技术。

OLE 技术是 ActiveX 技术的坚实基础，它的飞速发展使 ActiveX 技术逐渐浮出水面。在 1996 年 3 月的 Internet 专业开发人员研讨会（Internet PDC）上，微软公司最早提出 ActiveX 一词，当时 ActiveX 指的是大会的口号“Active the Internet”，这仅仅是一种号召，而不是具体的应用程序开发技术或体系结构。

此后，ActiveX 逐渐成为微软新的企业口号，就像 90 年代初期的 OLE 一样，而且在很短的时间内，它的含义就远远超过了“Active the Internet”。ActiveX 成了定义从 Web 页面到 OLE 的所有内容的核心术语。一方面，它表示将你联系到 Microsoft、Internet 和软件新技术的小型快速可重用组件；另一方面 ActiveX 代表了 Internet 与应用程序的一种集成策略。

如今，如果哪个产品或公司在它的词汇中不出现 ActiveX，那么不论从哪方面看，它都是落伍了。事实上，ActiveX 已经不仅仅是一种技术或一种体系，而是一种概念和潮流。

1.2 ActiveX、OLE 和 Internet

现在，ActiveX 和 OLE 已经成为了同义词，以前人们所说的 OLE 控件（Ocx）现在称为 ActiveX 控件，OLE DocObject 现在称为 ActiveX 文件。在许多情况下，有关如何实现 OLE 技术的文章已经被更新为 ActiveX 技术，并且仅仅是替换了 OLE 一词，它目前被称为 ActiveX。

虽然 OLE 和 ActiveX 技术已有了巨大的发展，而且看上去每天都有新技术在不断涌现，但是可以确定，这些都是由于 Internet 引发和推动的，Internet 的主要贡献在于对 Web 页面的实现和发布的需求。事实上，ActiveX 的每一个新特征都可以追溯到对小型快速可重用组件的需求，而这些都是以 OLE 和 COM 为基础的。ActiveX 并不是为了替换 OLE，而是将它进行扩展以适应 Internet、Intranet、商业应用程序以及家用应用程序的开发。

除了生成 ActiveX 技术的特定技术和开发工具以外，微软还建立了一套使用和集成 ActiveX 组件的标准，从 VB 到 Word，以及 Java 等所有产品都具有使用 ActiveX 组件的能力，使它们可以被组织到一起协同工作。而在几年以前，想把几个应用程序像今天这样紧密的结合到一起几乎是不可能的。