

Visual Basic 6.0 程序设计 — ActiveX 篇

刘炳文 编著

人民邮电出版社

内 容 提 要

Visual Basic 6.0 中文版是 Microsoft 公司推出的最新 Visual Basic 版本, 它简单易学、功能强大, 深受广大计算机专业人员和非专业人员的欢迎。本书介绍了 Visual Basic 6.0 中与 ActiveX 技术有关的一些功能, 主要内容包括: 对象和集合、Windows API、Visual Basic 提供的 ActiveX 控件、建立 ActiveX 控件和其它组件 (包括 ActiveX DLL、ActiveX EXE 以及 ActiveX 文档)、超文本标记语言 (HTML)、脚本编写语言 VBScript、可视化页面设计工具 ActiveX Control Pad、动态 HTML、多媒体程序设计以及 Internet 控件等。

本书既可供具有一定 Visual Basic 程序设计基础的读者自学, 也可作为大专院校、培训班的教材和参考书。

计算机实用软件丛书

Visual Basic 6.0 程序设计——ActiveX 篇

- ◆ 编 著 刘炳文
责任编辑 赵桂珍
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京顺义向阳胶印厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787 × 1092 1/16
印张: 36
字数: 902 千字 1999 年 12 月第 1 版
印数: 1 - 6 000 册 1999 年 12 月北京第 1 次印刷

ISBN 7-115-08056-9/TP·1275

定价: 52.00 元

《计算机实用软件丛书》编委会

高级顾问： 张效祥 胡启恒

主 任： 杜肤生

副 主 任： 徐修存 李树岭 罗晓沛

特约编委： 谭浩强 陈树楷

编 委：（按姓氏笔画排序）

王亚明 毛 波 方 裕 史美林

孙家骥 刘炳文 刘德贵 吴文虎

张国锋 周山芙 周堤基 钟玉琢

柳克俊 侯炳辉 赵桂珍 聂元铭

徐国平 寇国华 戴国忠

丛书前言

随着计算机、通信和信息技术的迅速发展与广泛应用,人类正在进入信息化社会。计算机技术的应用与推广,将直接推动社会信息化的发展;而计算机技术的应用与推广,实质上取决于计算机软件的应用和推广,可以说,没有软件,就没有计算机的应用;学习、使用计算机,从根本上讲就是学习和掌握软件的使用。

为了适应当前计算机技术发展的需要,满足读者学习、使用计算机软件的需求,人民邮电出版社约请有关专家编写出版了这套“计算机实用软件丛书”。

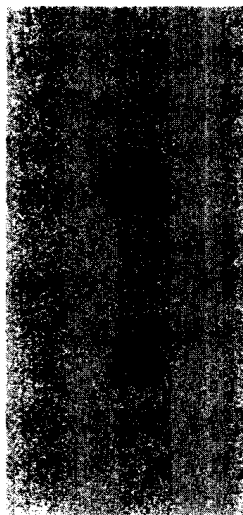
这套丛书的特点是:普及兼顾提高,应用兼顾开发,各书独立成册形成系列,并注重其相关性,使丛书成为广大计算机应用和开发人员学习使用计算机的必备用书。

这套丛书的内容包括:程序设计语言、操作系统技术、数据库技术、软件开发技术及工具、网络技术、多媒体技术等。

在计算机技术飞速发展的今天,软件产品更新快,经常有新产品或新版本问世,因此我们不但介绍当前流行和优秀的软件,而且力求尽快把国内外最新的软件产品也介绍给读者。

我们将全心全意为读者服务,也热切期待广大读者对丛书提出宝贵意见,以进一步提高丛书的质量。让我们共同努力,为提高我国的计算机开发、应用水平做出贡献。

“计算机实用软件丛书”编委会



Visual Basic是Microsoft公司最成功的编程语言产品，在全世界拥有数以百万计的用户。1991年推出后，由于界面友好、简单易学、功能强大，因而得到了迅速推广。随着版本的更新，Visual Basic已成为真正专业化的大型开发语言和环境，不仅功能越来越强，而且更容易使用。

Visual Basic 6.0是在Visual Basic 5.0的基础上推出的，在某些方面较5.0版有重要的改进，它所提供的开发环境与Windows 9x或Windows NT 4.0或5.0具有完全一致的界面，使用更方便，其代码效率已达到Visual C++的水平。在面向对象程序设计方面，6.0版的Visual Basic全面支持面向对象的程序设计，包括数据抽象、封装、对象与属性、类与成员、继承和多态等。

近年来，有一种技术正在迅速影响到现代生活的各个方面，这就是Internet。从某种程度上说，Web也许是有史以来最重要的一次通信革命，Microsoft的“Windows平台”正在受到“Internet平台”的挑战。为了保住自己的“霸主”地位，Microsoft推出了ActiveX技术。

ActiveX是Microsoft的Internet长远战略的基石之一。Microsoft试图通过ActiveX把Internet网或World Wide Web网与桌面（包括Windows桌面和Macintosh桌面）乃至UNIX合并到一起，组成一个“无缝”的集成化环境，从而使Internet可以像普通的硬件或局域网一样，成为从桌面访问的一个资源。也就是说，按照Microsoft的观点，不是把用户的桌面放到Internet上，而是把Internet放到用户的桌面上，而ActiveX就是实现这种观点的最重要的技术。

ActiveX技术体现在Microsoft的各种开发工具中，而在Visual Basic中体现得尤其充分。6.0版的Visual Basic提供了开发Internet应用程序的多种手段，可以在Visual Basic的集成环境下，利用Visual Basic提供的强大功能，开发能在Internet上运行的应用程序。对于已熟悉Visual Basic的用户来说，掌握ActiveX技术和用Visual Basic开发Internet应用程序都是很容易的事。

为了开拓Visual Basic的应用领域，满足读者了解ActiveX和用Visual Basic开发Internet应用程序的强烈愿望，笔者编写了这本《Visual Basic 6.0程序设计—ActiveX篇》。需要说明的是，本书所介绍的不是Visual Basic的基

基础知识，而是为有一定Visual Basic程序设计经验的读者编写的。对于初学者来说，可以阅读由人民邮电出版社出版的《Visual Basic程序设计—基础篇》，该书介绍了Visual Basic程序设计的基础知识，主要内容包括：Visual Basic的编程环境、基本语法、常用内部控件、程序控制结构、数组、过程、菜单程序设计、对话框程序设计、高级事件过程、多重窗体与屏幕对象、多重文档界面（MDI）、工程管理、环境应用以及程序调试、文件处理等。

《Visual Basic 6.0程序设计—ActiveX篇》是在《Visual Basic程序设计—基础篇》的基础上编写的，主要讲述如何使用Visual Basic 6.0的ActiveX控件、建立用户自己的控件和其它组件以及如何用Visual Basic开发Internet应用程序。全书共十四章，各章内容如下：

第一章到第三章是学习本书的预备知识，主要讲述了什么是ActiveX技术、如何在Visual Basic中定义和使用类以及如何使用Windows应用程序接口（API）。这些知识是学习本书其它各章必备的基础，应注意掌握。

ActiveX控件是Visual Basic的重要内容，也是ActiveX技术的组成部分。第四到第六章介绍了Visual Basic中的部分常用ActiveX控件。在介绍多媒体控件（第六章）时，同时介绍了Visual Basic的多媒体程序设计方法。

第七、第八章介绍了如何在Visual Basic中建立ActiveX组件，包括ActiveX DLL、ActiveX EXE和ActiveX控件。

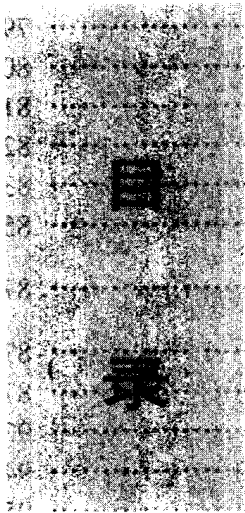
第九到第十一章介绍了Web页面设计技术，包括超文本标记语言（HTML）、脚本编写语言VBScript以及可视化页面设计工具ActiveX Control Pad。

6.0版的Visual Basic提供了建立Web页面的多种手段，可以方便地进行Internet应用程序设计。第十二和第十三章介绍了在Visual Basic中如何通过ActiveX文档和动态HTML建立Web页面。

Visual Basic提供了三个与Internet有关的ActiveX控件，可以用来实现各层的Internet通信。第十四章介绍了如何用Visual Basic的Internet控件访问Internet和进行通信。

目前，Visual Basic的应用正在向“纵深”发展。随着因特网的迅速普及，6.0版的Visual Basic增加了许多新的与Internet有关的内容，本书介绍的只是其中的一部分，希望它能在帮助读者用Visual Basic设计Internet应用程序方面起到一些作用。但是，由于笔者水平有限，经验不多，缺点和错误在所难免，诚望专家和广大读者不吝赐教，批评指正。

刘炳文



| | |
|---|----|
| 第一章 ActiveX技术 | 1 |
| 1.1 从剪贴板到ActiveX | 1 |
| 1.2 ActiveX技术 | 3 |
| 1.2.1 ActiveX Internet模型 | 4 |
| 1.2.1 ActiveX模型 | 5 |
| 1.3 ActiveX与其它技术 | 8 |
| 1.3.1 ActiveX与COM | 8 |
| 1.3.2 ActiveX与Java | 9 |
| 1.4 使用ActiveX的必要性 | 11 |
| 第二章 对象、类和集合 | 13 |
| 2.1 Visual Basic中预定义的和对象 | 13 |
| 2.1.1 预定义对象和类 | 13 |
| 2.1.2 对象的操作 | 15 |
| 2.2 对象变量 | 18 |
| 2.2.1 对象变量的声明和赋值 | 18 |
| 2.2.2 继承 | 22 |
| 2.2.3 对象变量的释放 | 26 |
| 2.3 属性 (Property) 过程 | 27 |
| 2.3.1 属性过程的定义 | 27 |
| 2.3.2 属性过程举例 | 31 |
| 2.4 初始化事件和终止处理事件 | 35 |
| 2.4.1 Initialize事件和Terminate事件 | 35 |
| 2.4.2 程序举例 | 36 |
| 2.5 建立自己的对象 | 37 |
| 2.5.1 类模块的基本概念 | 37 |
| 2.5.2 类的建立 | 40 |
| 2.6 集合 | 48 |
| 第三章 Windows应用程序接口 (API) | 55 |
| 3.1 静态链接与动态链接库 | 55 |
| 3.1.1 静态链接 | 55 |
| 3.1.2 动态链接库 | 56 |
| 3.2 在Visual Basic中使用动态链接库 | 57 |
| 3.2.1 声明 | 57 |
| 3.2.2 Visual Basic 6.0中的DLL过程声明 | 60 |
| 3.3 API文本浏览器 | 62 |
| 3.3.1 API浏览器的使用 | 62 |
| 3.3.2 把声明、常量或类型复制到Visual Basic代码中 | 66 |
| 3.4 API过程调用 | 68 |
| 3.4.1 调用API绘图过程 | 68 |

| | | |
|------------|--|------------|
| 3.4.2 | 文本输出 | 71 |
| 3.4.3 | 环境设置 | 72 |
| 3.5 | 句柄 | 75 |
| 3.5.1 | 窗口句柄 | 75 |
| 3.5.2 | 设备描述表句柄 | 79 |
| 3.6 | 特殊类型数据的传送 | 80 |
| 3.6.1 | 字符串的传送 | 81 |
| 3.6.2 | 数组的传送 | 82 |
| 3.6.3 | 其它数据的传送 | 83 |
| 3.6.4 | 把C文档中的声明转换为Visual Basic中的声明 | 85 |
| 第四章 | 常用ActiveX控件 | 87 |
| 4.1 | ActiveX控件与可插入对象 | 87 |
| 4.1.1 | ActiveX控件 | 87 |
| 4.1.2 | 可插入对象 | 92 |
| 4.2 | 网格 | 94 |
| 4.2.1 | 网格的属性 | 95 |
| 4.2.2 | 网格事件 | 101 |
| 4.3 | 三维控件 | 104 |
| 4.3.1 | 三维命令按钮控件 (3D Command Button Control) | 104 |
| 4.3.2 | 三维组按钮控件 (3D Group Push Button Control) | 106 |
| 4.3.3 | 三维面板控件 (3D Panel Control) | 108 |
| 4.4 | 量度表控件 | 112 |
| 4.4.1 | 量度表控件属性 | 112 |
| 4.4.2 | 程序举例 | 114 |
| 4.5 | 旋钮控件 | 118 |
| 4.5.1 | 事件和属性 | 118 |
| 4.5.2 | 程序举例 | 119 |
| 4.6 | 格式文本框控件 | 121 |
| 4.6.1 | 文件的装入和保存 | 122 |
| 4.6.2 | 字体设置 | 127 |
| 4.6.3 | 颜色设置 | 128 |
| 4.6.4 | 对齐与缩进 | 129 |
| 4.6.5 | 打印 | 132 |
| 第五章 | Windows公共控件 | 135 |
| 5.1 | 概述 | 135 |
| 5.2 | 滑动器 | 138 |
| 5.2.1 | 设置滑动器的外观 | 138 |
| 5.2.2 | 滑动器的操作 | 140 |
| 5.3 | 状态条 | 144 |
| 5.3.1 | 建立状态条 | 144 |
| 5.3.2 | 面板对象的操作 | 151 |
| 5.4 | 进度条 | 154 |

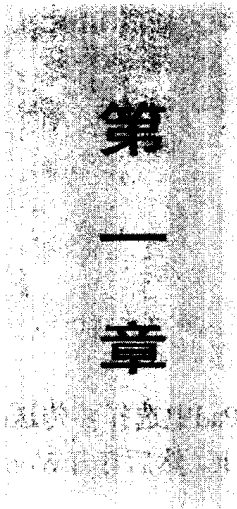
| | |
|--|------------|
| 5.4.1 进度条的显示位置 | 154 |
| 5.4.2 程序举例 | 155 |
| 5.5 工具条 | 157 |
| 5.5.1 工具条的位置与按钮对象 | 157 |
| 5.5.2 在工具条上添加按钮对象 | 158 |
| 5.5.3 按钮对象的大小和操作 | 163 |
| 5.5.4 显示工具提示文本 (ToolTips) | 164 |
| 第六章 多媒体控件和多媒体ActiveX EXE组件 | 167 |
| 6.1 概述 | 167 |
| 6.1.1 什么是多媒体和多媒体系统 | 167 |
| 6.1.2 多媒体元素 | 169 |
| 6.1.3 Windows的多媒体设置 | 171 |
| 6.2 Visual Basic的多媒体程序设计 | 176 |
| 6.2.1 Windows高级多媒体服务 | 176 |
| 6.2.2 Visual Basic的多媒体支持 | 177 |
| 6.3 MCI控件 | 178 |
| 6.3.1 MCI控件的基本功能 | 179 |
| 6.3.2 MCI命令及其使用 | 180 |
| 6.4 MCI控件的属性和事件 | 182 |
| 6.4.1 MCI控件属性 | 182 |
| 6.4.2 MCI控件事件 | 199 |
| 6.5 MCI控件程序举例 | 201 |
| 6.6 API多媒体函数 | 210 |
| 6.6.1 与多媒体有关的高级接口函数 | 211 |
| 6.6.2 高级接口函数的使用 | 212 |
| 6.7 MCI指令的构成和使用 | 214 |
| 6.7.1 MCI指令的构成 | 214 |
| 6.7.2 MCI指令的使用 | 216 |
| 6.8 部分MCI指令详解 | 219 |
| 6.8.1 MCI系统指令 | 220 |
| 6.8.2 MCI音频指令 | 221 |
| 6.8.3 MCI AVI指令 | 230 |
| 6.9 MCIWnd控件 | 240 |
| 第七章 建立ActiveX DLL和ActiveX EXE组件 | 245 |
| 7.1 ActiveX组件 | 245 |
| 7.1.1 基本概念 | 246 |
| 7.1.2 在建立组件时应注意的几个问题 | 247 |
| 7.2 ActiveX EXE组件 | 250 |
| 7.2.1 建立ActiveX EXE | 250 |
| 7.2.2 在其它应用程序中使用ActiveX EXE组件 | 254 |
| 7.3 建立含有窗体的组件 | 256 |
| 7.4 ActiveX DLL组件 | 260 |

| | |
|-------------------------------|------------|
| 7.4.1 建立ActiveX DLL组件 | 260 |
| 7.4.2 测试DLL | 263 |
| 第八章 建立ActiveX控件 | 267 |
| 8.1 引例 | 267 |
| 8.1.1 建立Visual Basic控件 | 268 |
| 8.1.2 建立ActiveX控件 | 274 |
| 8.2 基础知识 | 275 |
| 8.2.1 基本概念 | 275 |
| 8.2.2 UserControl对象 | 277 |
| 8.3 用向导为ActiveX控件添加属性 | 278 |
| 8.3.1 建立控件 | 279 |
| 8.3.2 添加属性 | 281 |
| 8.3.3 测试用户控件 | 289 |
| 8.4 属性页 | 291 |
| 8.4.1 建立属性页 | 292 |
| 8.4.2 设计属性页 | 294 |
| 8.5 用户控件的方法和事件 | 299 |
| 8.5.1 为控件添加方法和事件 | 299 |
| 8.5.2 测试修改后的控件 | 302 |
| 第九章 超文本标记语言(HTML)..... | 305 |
| 9.1 Web与URL | 305 |
| 9.1.1 什么是Web | 305 |
| 9.1.2 什么是URL | 306 |
| 9.2 超文本标记语言(HTML)简介 | 308 |
| 9.2.1 HTML文档的基本结构 | 308 |
| 9.2.2 分行、居中与注释 | 311 |
| 9.3 标题与列表 | 312 |
| 9.3.1 标题 | 312 |
| 9.3.2 列表 | 314 |
| 9.4 表格与文本 | 318 |
| 9.4.1 表格 | 318 |
| 9.4.2 文本 | 322 |
| 9.5 输入标记与内部控件 | 325 |
| 9.5.1 输入标记 | 325 |
| 9.5.2 内部HTML控件 | 327 |
| 9.6 窗体与图像 | 333 |
| 9.6.1 窗体 | 333 |
| 9.6.2 图像 | 335 |
| 9.7 对象标记与ActiveX控件 | 337 |
| 9.7.1 对象标记 | 337 |
| 9.7.2 ActiveX控件 | 338 |
| 9.8 超链接 | 346 |

| | |
|---|-----|
| 第十章 脚本编写语言VBScript | 347 |
| 10.1 JavaScript与VBScript | 347 |
| 10.1.1 脚本编写语言 | 347 |
| 10.1.2 VBScript与JavaScript | 348 |
| 10.2 VBScript语言与Visual Basic | 352 |
| 10.2.1 VBScript与Visual Basic的主要区别 | 353 |
| 10.2.2 可以在VBScript中使用的Visual Basic关键字 | 355 |
| 10.3 HTML与VBScript | 361 |
| 10.3.1 脚本代码在HTML文档中的位置 | 361 |
| 10.3.2 模块 | 362 |
| 10.3.3 事件处理 | 363 |
| 10.4 脚本对象模型 | 366 |
| 10.4.1 脚本对象的属性 | 366 |
| 10.4.2 脚本对象的方法 | 368 |
| 10.5 Window对象 | 370 |
| 10.5.1 Window对象的主要属性 | 370 |
| 10.5.2 Window对象的方法 | 372 |
| 10.6 Document对象 | 376 |
| 10.6.1 Document对象的属性 | 377 |
| 10.6.2 Document对象的方法 | 380 |
| 10.6.3 文档的颜色设置 | 381 |
| 10.7 Document对象举例 | 383 |
| 10.8 其它对象 | 392 |
| 10.8.1 History对象 | 392 |
| 10.8.2 Navigator对象 | 394 |
| 10.8.3 Location对象和Link对象 | 396 |
| 10.9 菜单 | 398 |
| 第十一章 ActiveX Control Pad | 403 |
| 11.1 Control Pad的构成和安装 | 403 |
| 11.2 文本编辑器和对象编辑器 | 404 |
| 11.2.1 文本编辑器 | 404 |
| 11.2.2 对象编辑器 | 405 |
| 11.3 布局编辑器 | 410 |
| 11.3.1 HTML布局编辑器的构成 | 410 |
| 11.3.2 控件的基本操作 | 413 |
| 11.4 建立.alx文件 | 415 |
| 11.4.1 设计界面 | 415 |
| 11.4.2 编写代码 | 416 |
| 11.5 建立HTML文档 | 419 |
| 第十二章 ActiveX文档 | 425 |
| 12.1 概述 | 425 |

| | | |
|------------------------------|--|------------|
| 12.1.1 | 什么是ActiveX文档 | 425 |
| 12.1.2 | 为什么要使用ActiveX文档 | 427 |
| 12.2 | ActiveX文档的构成 | 428 |
| 12.2.1 | 把现有的应用程序转换为ActiveX文档 | 428 |
| 12.2.2 | ActiveX文档的组成 | 433 |
| 12.3 | UserDocument对象和HyperLink对象 | 436 |
| 12.3.1 | UserDocument对象的主要事件 | 436 |
| 12.3.2 | HyperLink对象 | 440 |
| 12.4 | 建立ActiveX文档 | 441 |
| 12.4.1 | 建立两个ActiveX文档 | 441 |
| 12.4.2 | 向文档工程中添加窗体 | 446 |
| 12.5 | 文档的公有属性 | 449 |
| 12.5.1 | 添加属性 | 449 |
| 12.5.2 | 属性的保存 | 455 |
| 12.6 | ActiveX文档中的菜单 | 457 |
| 12.7 | ActiveX文档应用程序打包 | 462 |
| 12.7.1 | 应用程序打包 | 462 |
| 12.7.2 | 用“打包和展开向导”为应用程序打包 | 463 |
| 第十三章 动态HTML | | 479 |
| 13.1 | 概述 | 479 |
| 13.1.1 | 什么是DHTML | 480 |
| 13.1.2 | 为什么要使用DHTML | 481 |
| 13.2 | DHTML工程的编程环境 | 482 |
| 13.2.1 | 启动DHTML工程 | 482 |
| 13.2.2 | 设计器与工具箱 | 483 |
| 13.3 | 建立DHTML应用程序 | 487 |
| 13.3.1 | 工具栏 | 487 |
| 13.3.2 | 建立DHTML应用程序 | 492 |
| 13.4 | DHTML页面的保存 | 495 |
| 13.5 | 动态HTML中的属性和事件 | 499 |
| 13.5.1 | 设置和获取Web页面的属性 | 499 |
| 13.5.2 | 动态HTML中的事件 | 501 |
| 13.6 | 超链接与漫游 | 504 |
| 13.6.1 | 在页面设计器中建立超链接 | 504 |
| 13.6.2 | 在DHTML应用程序中漫游 | 505 |
| 13.6.3 | 部署DHTML应用程序 | 506 |
| 13.7 | DHTML应用程序举例 | 507 |
| 第十四章 Internet控件 | | 517 |
| 14.1 | WebBrowser控件与InternetExplorer对象 | 517 |
| 14.1.1 | WebBrowser控件与InternetExplorer对象的引用 | 518 |
| 14.1.2 | WebBrowser控件与InternetExplorer对象的属性 | 519 |
| 14.1.3 | WebBrowser控件与InternetExplorer对象的方法 | 520 |

| | | |
|--------|--|-----|
| 14.1.4 | WebBrowser控件与InternetExplorer对象的事件 | 522 |
| 14.2 | WebBrowser控件举例 | 524 |
| 14.3 | WinSock控件 | 534 |
| 14.3.1 | WinSock控件的属性 | 534 |
| 14.3.2 | WinSock控件的方法 | 537 |
| 14.3.3 | WinSock控件的事件 | 539 |
| 14.4 | 用WinSock控件建立“聊天”程序（服务器端） | 542 |
| 14.4.1 | TCP与UDP | 542 |
| 14.4.2 | 建立服务器端应用程序 | 543 |
| 14.5 | 用WinSock控件建立“聊天”程序（客户端） | 555 |



第一章

ActiveX 技术

ActiveX是近年来新升起的一颗“网络新星”，它是Microsoft的Internet长远战略的基石之一。Microsoft试图通过ActiveX把Internet网或World Wide Web网与桌面（包括Windows桌面和Macintosh桌面）乃至UNIX合并到一起，组成一个“无缝”的集成化环境，从而使Internet可以像普通的硬件或局域网一样，成为从桌面访问的一个资源。

随着Web的发展和开发，ActiveX技术也在不断更新和进步。在这一章中，我们将对ActiveX及其相关技术作一个简单介绍，使大家对ActiveX有一个总体印象。

1.1 从剪贴板到ActiveX

不少人对ActiveX还不很了解，但对剪贴板想必非常熟悉，而且经常使用。实际上，ActiveX技术就是从剪贴板发展而来的。这一节将介绍ActiveX的产生过程。

在实际应用中，常常需要在两个应用程序之间交换信息。DOS环境下的信息交换比较繁琐，往往要通过中间文件来实现。而在Windows中，应用程序之间的信息交换大致经历了以下几个阶段：

1. 剪贴板

Windows环境下信息交换最原始的方式是通过剪贴板（Clipboard）来实现的。剪贴板是一个临时的存储区，为了把文本或图形从一个应用程序（源）复制到另一个应用程序（目标），可以先把要复制的内容放到剪贴板中，然后再“粘贴”（Paste）

到目标处，如图1.1所示。

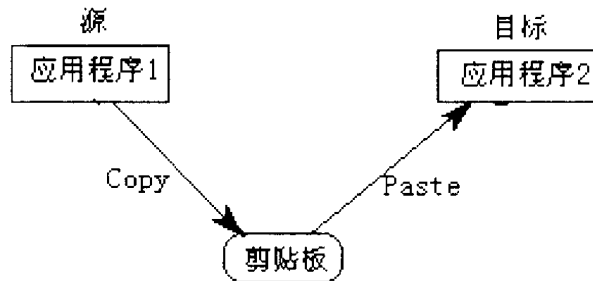


图1.1 通过剪贴板在两个应用程序之间传送数据

例如，为了把NotePad中指定的文本传送到Word中，可以先在NotePad中选择要传送的文本，并执行“编辑”菜单中的“复制”命令，把这部分文本放到剪贴板上，然后打开Word，执行“编辑”菜单中的“粘贴”命令，即可完成。

2. 动态数据交换 (DDE)

用剪贴板传送数据的操作比较简单，但需要对数据进行手工复制（或剪切）和粘贴。为了能直接把数据从一个应用程序传送到另一个应用程序，Microsoft公司又引入了动态数据交换（DDE）协议。利用DDE，可以通过程序代码使两个应用程序之间进行通信。在使用DDE的两个应用程序中，其中一个为信息接收方，称为客户，另一个为信息提供方，称为服务器，如图1.2所示。

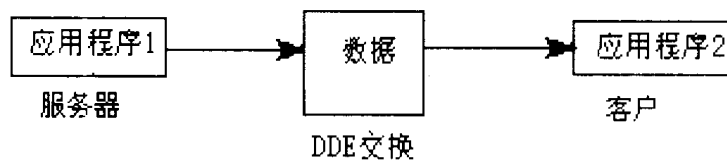


图1.2 通过DDE进行通信

3. 对象链接与嵌入 (OLE)

DDE在两个应用程序之间提供了强有力的通信协议，但是，编写和调试基于DDE的程序是一件相当困难的事情。为此，Microsoft公司又开发出了对象链接与嵌入（OLE）技术，它通过一种“以文档为中心”的计算模型，使得一个应用程序可以使用其它应用程序的功能。例如，通过OLE，可以在Word文档中插入一个Excel工作表，并可使用Excel的所有功能。也就是说，如果在编辑Word文档时需要修改工作表，则可以通过双击工作表启动Excel，对该表进行修改，完成修改后退出Excel，并返回到Word。

一般来说，当在文档中嵌入了一个诸如Excel这样的对象时，如果希望能对该对象进行修改，则要求系统中有相应的软件。而在今后，文档中的对象将带有自己的功能，无需接收方的系统中有相应的软件。

在使用OLE时，既可以在文档中嵌入一个对象，也可以在其中链接一个对象。当嵌入

一个对象时，将在文档中存放一个此对象的复制，如果以后在文档外对此对象进行了修改，则文档中的对象将不会受到影响；而如果在文档中链接了一个对象，则只在文档中存放一个对象文件的索引，每次打开文档时，都要把该对象相应的文件装入文档中，因此，如果修改了对象，则可在文档中保持该对象的最新内容。OLE容器（客户）是一个嵌入或链接了对象的应用程序；而OLE服务器则是建立这个被链接或被嵌入对象的应用程序。例如，当把Excel工作表嵌入或链接到Word文档后，Word是客户，Excel为服务器，如图1.3所示。

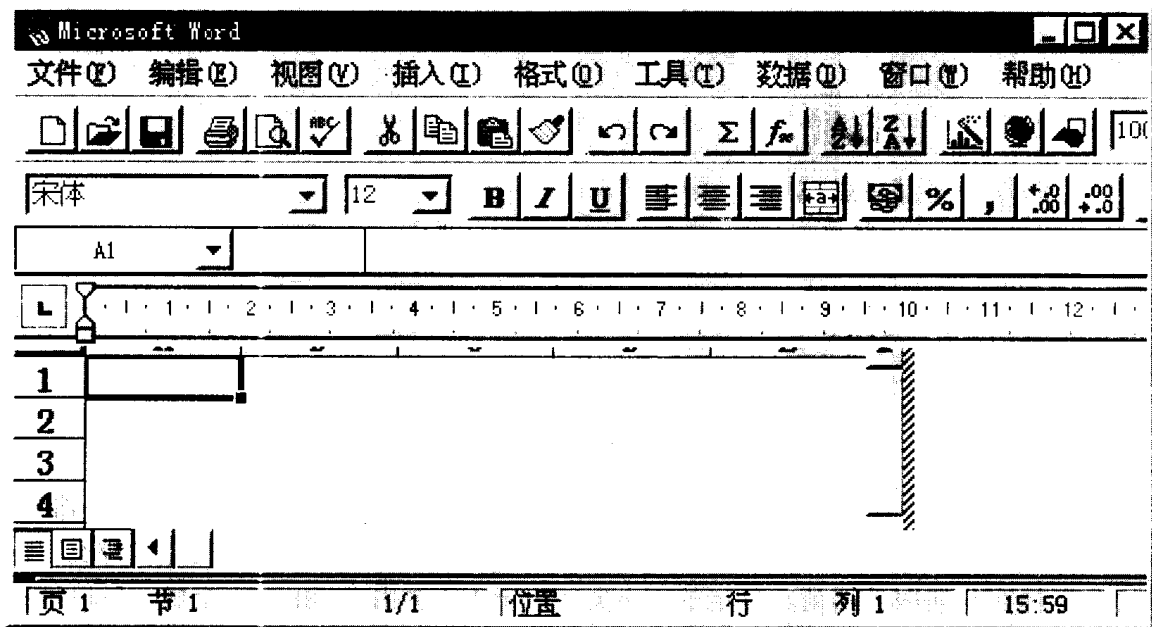


图1.3 在Word文档中嵌入Excel工作表

4. ActiveX

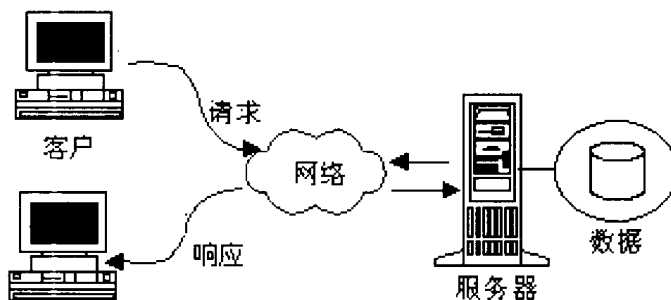
随着Internet的迅速发展及普及，人们把注意力转向了网络，需要开发出一种新的对象技术。在这种情况下，作为一种为网络定义的OLE，Microsoft ActiveX应运而生。现在我们似乎还无法给ActiveX下一个完整而明确的定义，只能把它看作是一种用于Internet的OLE技术。

1.2 ActiveX技术

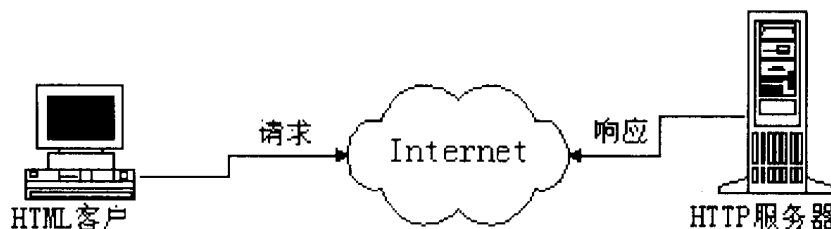
如前所述，ActiveX是一种基于Internet的技术。有了这种技术，就可以实现用户计算机和信息之间的无缝连接，而不必关心这些信息存放在什么地方。在这一节中，将介绍ActiveX Internet模型和ActiveX的构成。

1.2.1 ActiveX Internet 模型

早期的计算模型是基于主机的，用户通过终端来运行存放在主机中的程序，随着工作站和计算机网络的应用，出现了客户/服务器计算模型。在这种计算模型中，用户运行存放在自己PC机中的应用程序，并通过网络向服务器发送请求信息，在客户端主要处理用户界面及用户输入，而服务器则处理客户的请求，客户与服务器通过网络进行通信，如图1.4所示。



随着Internet的广泛应用，超文本传输协议（HTTP）模型继续发展了客户/服务器模型。在基于HTTP的模型中，客户端（通常为浏览器）与服务器进行交互，在一般情况下，这种交互包括客户向服务器请求指定的Web页。由于Web页由HTML文档组成，所以这种模型有时也称为HTTP：HTML客户/服务器模型，万维网就是基于HTTP的，如图1.5所示。



从用户的观点看，HTTP是非动态的，它只提供很少的交互甚至根本没有交互功能。也就是说，用户只能简单地查看Web页中的内容，而不能通过Web页中的元素进行交互。这种交互能力可以通过公共网关接口（CGI）、Perl及其它编程语言来实现。

利用HTML，可以建立一个窗体，这样，用户通过浏览器就能与服务器进行交互。当用户单击浏览器中窗体的Submit按钮时，浏览器将该窗体发送给服务器，服务器运行一个程序（通常用Perl编写）来处理所提交的窗体。根据服务器程序的设计，可能生成一个基于HTML的响应，并由服务器发送回客户浏览器。

基于CGI的窗体处理程序有一个缺点，就是每次当客户启动一个CGI脚本时，服务器都要建立一个新进程。随着进程的增多，服务器的处理能力会大为下降。使用CGI的客户/