

[美] W. 厄恩斯特

上网操作的
最直接方法
最关键技术

- 漫游网络的初级技术
- 漫游网络的软件工具
- 网络资源介绍
- 网络信息查询

Microsoft

ACTIVE X

技术与应用

科学出版社

Microsoft ActiveX 技术与应用

〔美〕 W. 厄恩斯特 著

韩国新 杨光 等译

韩国新 校



科学出版社

西蒙与舒斯特国际出版公司

1997

内 容 简 介

本书分五部分16章向读者展示了 Microsoft ActiveX 将静态的 Web 页面转换为生动的、交互性应用的技术内涵,介绍了如何利用 ActiveX 技术(诸如 ActiveX 控件、ActiveX 动画、ISAPI 过滤器及 VBScript 等)开发新的应用程序,还说明了 ActiveX 与 Java 的区别。

本书适用于对 ActiveX 技术感兴趣的读者。无论是新用户,项目策划者还是程序开发人员,本书将以其深入浅出的讲解,使读者了解 ActiveX 将带给 Web 的革命性变化的技术概念与细节。

Warren Ernst

PRESENTING ACTIVEX

Authorized translation from the English language edition

published by Sams. net Publishing

Copyright ©1996 by Sams. net Publishing

All rights reserved. For sale in P. R. China.

本书中文简体字版由科学出版社和美国西蒙与舒斯特国际出版公司联合出版。未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 PRENTICE HALL 防伪标签,无标签者不得销售。

版权所有,翻印必究。

图书在版编目(CIP)数据

Microsoft ActiveX 技术与应用/(美)厄恩斯特(Ernst, W.)著;
韩国新等译. -北京:科学出版社,1997. 6

ISBN 7-03-005719-8

I. Mic… II. ①厄…②韩… III. 全球网络:互连网络-接口-软件包, ActiveX N. TP317

中国版本图书馆 CIP 数据核字(96)第19642号

科学出版社 出版
西蒙与舒斯特国际出版公司

北京东黄城根北街16号

邮政编码:100717

北京双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1997年7月第 一 版 开本:787×1092 1/16

1997年7月第一次印刷 印张:11 1/4

印数:1-5000 字数:268 200

定价:18.00 元

译者的话

本书是 ActiveX 基础知识的介绍,其中包括:ActiveX 如何类似和区别于 Java 技术,如何运行 ActiveX 程序和控件,ActiveX 如何工作于 Internet 或非 Internet 环境,以及 ActiveX 技术的基本构成等等。本书的组织安排可以使具备广泛技术背景的用户对 ActiveX 本质有一个较好的理解。

书中首先讨论了 ActiveX 的意义,以及它对 Internet 和程序设计所带来影响。从中你可以了解到有关微软 Internet Explorer 3.0 Web 浏览器的概貌,包括如何从 Internet 上获取、安装和开始使用 ActiveX 技术。然后,通过简单的语言和大量的示例、屏幕画面、图表、少量代码和技术术语阐述了 ActiveX 不同方面的主要特性。接着,介绍了使用 Visual Basic 和 Visual C++ 创建 ActiveX 控件的过程和方法。在本书的最后讨论了一些 ActiveX 问题和前景。

本书不仅可作为那些项目计划制订者、软件开发人员和 Web 管理员的快速入门读物,同时也提供了有关介绍资料和一些基本的编程、代码示例。

本书是北京彤云软件创作室集体工作的结果,直接参加本书翻译工作的有韩国新、彤云、周艳、赵纬、周磊和韩国英。在翻译过程中,我们本着忠实原著、通俗易懂的原则,精心地选择了每一个术语,但由于水平有限,缺陷和错误在所难免,敬请读者批评指正,谢谢!

译者

1997年3月于北京

目 录

引 言	1
第一部分 Internet 和 ActiveX 基础	
1 今天的计算和 Internet	5
连接与计算	5
“连接”和“计算”是不同的领域	9
2 微软的 Internet 未来蓝图	11
结合连接与计算	11
ActiveX 与 Java 的区别	12
ActiveX 为 Internet 和 Web 带来的改变	13
后续内容	14
3 Internet Explorer 3.0: 迈向 Internet 未来的微软通途	15
Internet Explorer 3.0 起步	15
使用 Explorer	19
后续内容	22
第二部分 ActiveX 组件	
4 ActiveX 组件	25
ActiveX 控件: ActiveX 的基础	25
用 Visual Basic Script 撰写 ActiveX 脚本	28
把“通用程序和文件”书写为 ActiveX 文档	29
ISAPI 过滤器: 用于 Web 服务器的 ActiveX 技术	30
后续内容	31
5 ActiveX 文档	32
新信息和新文件格式的问题	33
ActiveX 文档: 最佳选择	36
后续内容	39
6 ActiveX 控件	40
ActiveX 控件及其优点	40
从头开始创建 ActiveX 控件	42
在 Web 页中使用 ActiveX 控件	43
ActiveX 控件的功能	43
后续内容	45
7 ActiveX 脚本	46
Web 页脚本	46
ActiveX 的 VBScript	48
后续内容	52

8 ActiveX Internet 信息服务器	53
ActiveX Web 服务器带来的功能和简便易用性	53
CGI 和 ISAPI 的区别	55
构造 ISAPI 过滤器	56
后续内容	57
第三部分 开发 ActiveX 控件	
9 ActiveX 控件和 VBScript	61
在 Web 页中嵌入 ActiveX 对象	64
VBScript 简介	70
VBScript 和 HTML 表单	73
VBScript 和 ActiveX 控件	75
后续内容	77
10 使用 ActiveMovie 控件	78
ActiveMovie Stream Editor	80
Web 中的 ActiveMovie	85
后续内容	93
11 ActiveX DocObject 和 Internet Explorer: 使你的应用支持 Internet 和 Intranet	94
DocObject 技术	95
ActiveX 和企业	95
ActiveX DocObject 及其历史的深入探讨	96
一个 ActiveX DocObject 的例子	101
实现想法: 如何利用这种技术	111
小结	111
12 用 ActiveX ISAPI 过滤器控制 Internet 信息服务器	112
ActiveX ISAPI 过滤器技术	112
ActiveX 与商业	113
事件通知变换表	117
用于 URL 映射的 ActiveX ISAPI 过滤器	119
实现思想: 如何使用这种技术	124
小结	125
后续内容	126
13 用 ISAPI 扩展开发 Web 应用	127
Web 服务器和 CGI 的特性	127
CGI 脚本的工作方式	128
CGI 的微软对策: ISAPI 扩展	129
扩展 API 集和相关的数据结构	130
REGBROWSE: 一个服务器目录浏览器	135
调试 ISA 扩展	141
后续内容	142

第四部分 ActiveX 的未来

14 ActiveX 应用	147
ActiveX 的联机多媒体	147
ActiveX 的 Web 应用	150
后续内容	153
15 ActiveX 和 Java	154
技术目标	154
ActiveX 和 Java 的实际结果	156
这一切意味着什么	157
小结	160
16 ActiveX 迈向 Nashville	161
ActiveX 的演变	161
Internet Explorer 3.0	163
Nashville	164
小结	166

第五部分 附 录

A ActiveX 联机资源	169
ActiveX 的正式信息资源	169
ActiveX 的非正式信息资源	170
一般的 Web/Internet 信息	170
B 术语	171
C VBScript 语法快速参考	176

引 言

就目前的状况而言, Internet 是一个不用考虑信息的实际存储位置, 用来搜索既存信息的杰出工具; 而商业应用则是擅长于为用户创造新信息的工具。你很少有机会找到既能完成信息产生, 又能利用 Internet 进行通信的“杂交”程序, 这主要归因于还不能广泛地获得用来构造这种应用的工具, 或者这种工具还不能强大到可以用来构造一个真正有用的运行顺畅程序。微软已经有了致力于改变这一状况的计划, 这就是 ActiveX 技术。

ActiveX 是一组软件技术和软件方法的总称, 其目标是使开发者可以很容易地创造出混合式的应用, 即带有强大的 Internet 互联支持的“传统商业应用”和带有真正计算能力的“传统 Internet 通信程序”。基于已经存在的 OLE (对象链接与嵌入) 技术, ActiveX 主要是通过把桌面系统“对象共享”的范围扩展到整个 Internet, 以使得构造这类新应用程序成为可能。由于 ActiveX 技术是一种设计标准, 因此, ActiveX 程序可以实现为一个独立的应用、Visual Basic 程序中的嵌入“智能”对象、Web 页或者是商业文档中的传统 OLE 对象, 所有这些都具备你所需要选择的 Internet 通信潜力。

随着 ActiveX 获得更加广泛的接受, Internet 将会从一个仅仅简单地提供静态信息“页”之源变为一个操纵、显示、修改和产生信息及数据的工作程序的“源泉”。因此在我们的文字处理器、电子表格和绘画等程序中就可以利用 Internet 进行通信, 而不需要转入到专用的浏览器或 Internet 程序中。

或者, 至少理论上如此。

作为一种非常新的技术, 骤然间, ActiveX 已经进入了 Internet 世界, 并且一夜之间改变了它。从已经开发完成的 ActiveX 技术来看, Internet 未来蓝图的实现将不会太遥远了。这就是本书所希望帮助你完成的: 研究 ActiveX 为软件开发者带来的可能性。

作为 ActiveX 的介绍, 本书特别适合于以下几类人:

- 需要知道 ActiveX 会对其产品或组织有何贡献的经理和计划制订者。
- 需要增加自己 Web 页吸引力和功能的 Web 页开发者。
- 需要了解 ActiveX 工作机制的程序员。
- 跟踪 Internet 技术前沿的爱好者。

如何使用本书

本书是 ActiveX 基础知识的介绍: ActiveX 如何类似和区别于 Java 技术, 如何运行 ActiveX 程序和模块, ActiveX 如何工作于 Internet 或非 Internet 环境, 以及 ActiveX 技术的基本构成。本书的组织安排可以使具备广泛技术背景的用户对 ActiveX 本质有一个较好的理解。

本书内容

本书首先讨论了 ActiveX 的意义,以及它对 Internet 和程序设计所带来影响。从中你可以了解到有关 Microsoft Internet Explorer 3.0 Web 浏览器的概貌,包括如何从 Internet 上获取、安装和开始使用 ActiveX 技术。然后,通过简单的语言和大量的示例、屏幕画面、图表和少量的代码和技术术语阐述了 ActiveX 不同方面的主要特性。接着,介绍了使用 Visual Basic 和 Visual C++ 创建 ActiveX 模块的过程和方法。在本书的最后讨论了一些 ActiveX 问题及其前景。

本书并不试图做为了一本详尽的编程指南或完整的 ActiveX 参考手册。实际上,本书是那些项目计划制订者、软件开发人员和 Web 管理员的快速入门读物,同时也提供了有关介绍资料和一些基本的编程、代码示例。

最后,本书是基于 1996 年初夏的 ActiveX 软件开发包编写的。随着时间的推移,某些方面可能已经过时或者需要进行改变才能反映实际情况。经常地浏览一下微软公司的主页 <http://www.microsoft.com/>,会得到有关 ActiveX 的新发展或下载 ActiveX SDK 的新内容。

本书的组织

本书分为五个部分(第五部分包括 4 个附录)。每部分包括 4 至 5 章,分别试图从不同的侧面来研究 ActiveX。尽管各部分之间的有些材料是反复出现的,但每一次都会有一个不同的“倾向”。

第一部分“Internet 和 ActiveX 基础”,是有关 Internet 以及 ActiveX 在 Internet 中的作用的概述。各章研究了目前 Internet 在传输非静态不变信息方面的不足、Internet 如何支持新技术以及如何用 Internet Explorer 3.0 来开始使用 ActiveX。

第二部分“ActiveX 组件”,定义和解释了 ActiveX 技术中的各个组成部分。每一章都针对一种 ActiveX 技术,解释它们与现存技术的异同点,介绍它们的创建和使用,同时也给出了供练习使用的示例。

第三部分“开发 ActiveX 控件”,比第二章更进一步,引导你经历创建不同 ActiveX 模块的过程,其中包括撰写脚本、预编译模块以及增强 Web 服务器等。

第四部分“ActiveX 的未来”,讨论了使用 ActiveX 的一些潜在的问题和解决方案,并把它们与 Java 技术的相关问题进行了比较,同时研究了如何可以把 ActiveX 技术与现行的 Windows NT 版本相结合。

附录总结了 Internet 上可获得的 ActiveX 资源,给出了术语定义,提供了 VBScript 的快速参考。

第一部分 Internet 和 ActiveX 基础

- 1 今天的计算和 Internet
- 2 微软的 Internet 未来蓝图
- 3 Internet Explorer 3.0: 迈向 Internet 未来的微软通途

1 今天的计算和 Internet

作为一个 Internet 用户，尽管你还没有意识到，Internet 实际已经处于一个细微的全面革新的边缘。它不仅会改变 Internet 的设计和构造，而且也会改变你的使用方式。

通过浏览器搜索到的包含 Java applet 和 JavaScript 形式的 Web 页，你实际已经感受到了这种未来的潜力。这些 Web 页暂时使你的浏览器具备了以前不曾有过的特征：智能（至少是通过一种有限的、暂时的表现方式）。“智能”一词，是指你的浏览器不仅可以显示静态、不动、不变的 Web 页，而且可以用一种新的、“智能”的方式来展示信息。

微软公司希望能推动这场 Internet 革命，这场革命不仅可以给浏览器增添智能，而且它也可以给“传统的商业应用”（比如文字处理器、电子表格和数据库程序）增添浏览器一样的连接性。微软计划通过广泛采用一系列称为 ActiveX 的技术来实现这一目标。ActiveX 是一种编程和 API（应用程序接口或程序之间相互通信的方法）的协议，是通过简单的修改现存程序而产生真正革命化结果的微软技术的演化。

为了理解结合当前“Internet”和“传统商业”应用潜在的好处，你需要了解正在使用的 Internet（及用来与其交互的程序）和商业程序以及二者之间的异同点。

连接与计算

表面上，电子表格程序和 Web 浏览器的操作是相似的。例如在 Windows 95 中，Excel 和 Netscape 都从开始菜单的图标中启动，都使用下拉菜单、对话框按钮和鼠标进行交互，都用相同的方式进行最小化、改变尺寸和关闭窗口。对于很多用户来讲，它们可能具有很类似的用途：用 Excel 来查看电子表格文件，用 Netscape 来浏览 Web 页。

尽管它们看起来或感觉起来是相似的，但这些应用具有非常明显的差别，每一个都具有其他所不具备的显著功能。例如，Excel 可以从现存的信息中计算新的数据（或者，至少对于知道如何使用的人来说是如此），Netscape 可以越过 Internet 获取远程信息。

难道你还认为这种差别还不足以促使我们来完成这种结合工作？不相信连接和计算之间真的存在着如此显著的差异？认为 ActiveX 对这些不同方面的结合并未带来革命？有很多人是这样认为的，但是回顾一下连接程序和计算程序之间存在的差异，将有助于对融合二者的作用有一个正确的看法。

连接：获取基于 Internet 的信息

现代 Web 浏览器，像 Netscape 或 Internet Explorer，是连接软件时代的杰出代表。它们被设计的可以操纵许多 Internet 协议及其两倍以上文件格式，所有这些功能都是

通过一个传输速率从每秒 1 千字节到每秒 10 兆位的连接所进行的,而且做起来似乎异常简单。即使是一个计算机或 Internet 新手,也可以使用这些程序来检索基于 Internet 的数据,而不需要任何指导——几乎完全就是靠鼠标点击来完成。

然而这一简单的界面下却隐含着非常复杂的程序结构,即最终的通信程序,它是一个 Web 浏览器的精髓。一个 Web 浏览器所做的全部就是与各种各样的 Internet 服务器进行通信,首先请求服务器应答建立基本连接,然后请求服务器传输信息,接着再把这些信息通过浏览器本身的简单的计算处理后显示给 Web 浏览器用户。当然,这是显示一个 Web 页过程的简单概述,通过进一步的研究,你会发现 Web 浏览器不仅仅是一个基于通信、面向图形的 Internet 对讲机。

下面是在你给出 URL 之后,Web 浏览器显示一个简单的 Web 页的所有步骤。不用担心,所有描述都仅使用简单的语言。请相信,这很有趣:

1. 浏览器查看 URL,对它进行分解。它断定,如果你给出了一个形如 `http://www.cris.com/~wernst/index.html` 的 URL,你是要使用超文本传输协议 (http) 来获得名为 `index.html` 的 Web 页,该页位于名为 `www.cris.com` 的 Internet 主机 (或服务器) 的 `~wernst` 子目录下。你会发现,在分解 URL 的过程中包含了少量的计算——如同你在语法学校做的那样,浏览器仅仅需要执行分解句子的工作。

2. 浏览器通过查看你的网络配置来查找最近的预定义的域名服务器 (DSN) Internet 地址,在 DSN 中有一个存储 Internet 服务器信息的大数据库或表格。浏览器把你要访问的服务器名称传递给 DSN (本例中的 `www.cris.com`),然后 DSN 通过 TCP 协议把该服务器的数字地址 (称为 IP 地址) 返回给浏览器 (在本例中是 `199.3.12.172`)。至此,访问了两个表格,浏览器和 DSN 进行了一些通信——但是,并没有很多真正的计算。

3. 然后,根据 URL 中的描述,浏览器从 DSN 取得地址信息,并利用 HTTP 协议和 IP 地址 `192.3.12.172` 与 `www.cris.com` 进行通信。这一初始通信联系可以分解为:浏览器的请求“嘿, `www.cris.com`,我是名为 Netscape (或 Internet Explorer 等) Web 的浏览器,想要检索你目录下的一些 Web 页。怎么样?你在工作吗?”;服务器的应答通常是“是的,你需要什么?”。至此,从头到尾,你已经增加了二行的通信和很少量的计算。

4. 现在,浏览器知道服务器正在工作,于是请求“我希望阅读你 `~wernst` 目录下的 `index.html` 文件”。服务器便搜索自己的目录,查找所要求的文件,查到后,利用一系列自己的表格来检查文件类型的合法性,然后发送消息“我找到了,给你”,并发送出该文件。如果服务器找不到所需要的文件,它就会回答“我没有你需要的文件”(或者给出其他出错信息,这依赖于不同系统的实现),然后终止通信。至此,你仅仅进行了更多的通信和表格查找,但并没有太多的计算。

5. 假设文件存在,浏览器就会以 HTML 文件的格式收到该 Web 页,然后开始对其进行解码。HTML 文件是人们可以阅读的文本文件,基本上不需要真正的计算来进行解码和显示。例如,当浏览器发现嵌入的图像后,它发送另一个信息给服务器:“等一等,在该页中我还需要文件 `image1.gif`,能马上传送给我吗?”服务器收到这一消息后,检查自己的文件目录和文件类型表,以确保能够传送浏览器请求的文件,然后回答“好”,并传送出内嵌的图像文件。对于 HTML 中的每一个嵌入的图像都要重复这一过程,直到浏

览器处理完所有的嵌入图像。随着文件的传送（通过大量的通信），浏览器以 HTML 文件中的代码为指导，利用收到的内嵌图像，在屏幕上构造该页内容，这一过程会利用 CPU 的部分计算能力来产生屏幕字体、颜色和图像。

图 1.1 用图形的方式表示了所有这些过程。

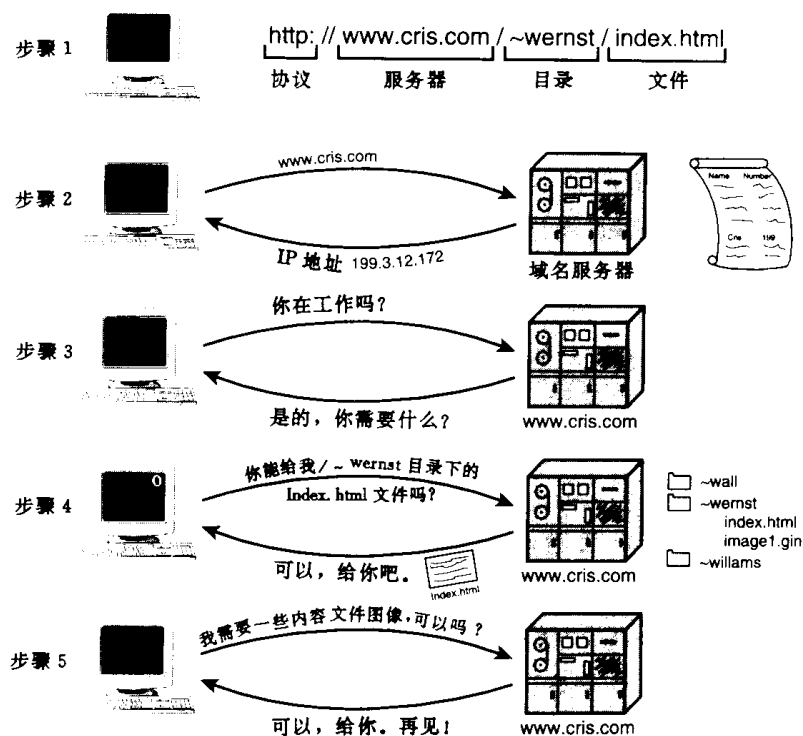


图 1.1 获得普通 Web 页的步骤

在本例中有很重要的几点需要说明。首先，传输一个 Web 页需要很多通信联系，不仅仅是与 URL 指定的 Web 服务器进行通信——该 Web 服务器的地址必须从域名服务器获得，而且需要使用一个与 Web 服务器进行通信的超文本传输协议完全不同的协议。

其次，Web 页的检索过程被分解为浏览器和 Web 服务器之间的一些较小的、具有不同目的的通信过程；它并非浏览器和 Web 服务器之间对 HTML 文件的一个突如其来的传输，尽管在 Web 漫游中你会有这样的感觉。

第三，在所有这些传输和显示的过程中只有很少量的计算。DSN 利用检索表来查找所请求的 Web 服务器 IP 地址，Web 服务器利用检索表来检查文件类型和目录定位，浏览器则利用检索表来确定用来显示 Web 页的字体和颜色。浏览器在显示 Web 页各部分的过程中没有丝毫的倾向和决策能力——它毫不越轨，简单地逐行通过 HTML 文件，按文件中指令行事。浏览器不需要，因此也就没有太多的计算能力或者说“智能”（我这样称呼它）。



我已经反复地使用了“检索表”一词，而没有解释使用它到底需要多么少的计算，或者给出其概念性的描述。我只想说，检索表可以想象成一个只有二列的查询表格，在第一列找到对应的条目，从第二列获得其相关的内容，然后用第二列的内容来完成一些工作。对于计算机来说，这是非常简单的事情。

最后，或许是最重要的一点，在这个过程中不会产生任何新的信息。所有的通信都被用来进行从一个结点（Web 服务器）到另一个结点（Web 浏览器）传递只用来查看的现存信息。这并不是说对于 Web 用户这些不是新信息，而是说在被浏览器收到之前，这些信息就已经存在于 Web 页上了，而且，它们在浏览器阅读的过程中不再改变。

表单 (Forms) 和 CGI 脚本 (Scripts) 怎么样呢？

前面段落的讨论描述了在 Web 浏览器中查看一个“正规的”Web 页的过程，其中没有包括需要填写的表单或动态改变的信息。也许你会问，Web 页的表单会增加 Web 页中的计算量吗？对于 Web 服务器来讲的确如此，但对于浏览器来说则绝对不会。



在大多数的 Web 页中，当按下 Submit Form（提交表单）按钮向 Web 服务器传送输入表单的信息时，浏览器的全部工作就是发出一个这样的消息“在你刚刚传送给我的 Web 页中，表单域已经被填写。如果你需要这些信息，那么请接收”，在从服务器收到“就绪”的消息后，它就把键入表单的信息原样发送出去，然后等待回答。

服务器收到信息后，对其进行分解，然后可能会运行一个实际的计算程序，在其他的表中查找一些信息，或者执行一个脚本（或执行其他许多计算新信息的方法中的一个）。服务器所真正要做的，就是计算出结果，用新产生的计算结果或信息构造一个 HTML 文件。然后，它回答“我有一个你需要显示的 HTML 文件。如果你能够接收，随后就是”。于是浏览器显示新的 HTML 文件，该文件的内容可能就是服务器计算的结果。

因此，就传统的 Web 页而言，浏览器的工作仅仅是与服务器进行通信和显示 Web 页：它并不实际完成很多其自身的计算。不依赖于浏览器的能力，服务器自己做了全部的计算（同时，或许会在计算的过程中产生新的信息）。

计算：在本地生成新信息

尽管本节的标题看起来相当高不可攀，在没有意识到的情况下，实际上你每天都在实践中用自己的计算机产生新信息。你也许会用 Microsoft Excel 来计算数据和画图表，

用 Microsoft Word 录入新数据表、数据字典及完整的著作，或者用 Microsoft Access 来录入、跟踪和查询数据——简言之，你在生成数据。

即使你用这些程序仅仅是为了查看其他人建立的数据，你还是具备从无到有或从硬盘的其他数据产生新数据的潜力，这是 Web 浏览器所不具备的（即使是像 Netscape Gold 这样的内嵌 HTML 编辑器的浏览器也一样）。更进一步的，这些程序工作于你的联机文件系统中——在大多数情况下，包括你的硬盘组和软盘驱动器——尽管也可能包括 Novell 这样的局域网环境。它们不能工作于你的联机系统之外。

事实上，对于一个面向软件或面向服务的行业，桌面应用的信息输出可能是其唯一的“产品”。对这部分人来讲，这些程序产生以前从未有过的信息，这一事实是非常明显的。

“连接”和“计算”是不同的领域

不论是偶然经历，还是通过亲自设计，大多数人认识到在连接软件和计算软件之间存在着根本区别，因此他们用不同的方式来使用这两类软件。这种现象没有什么奇怪，但它揭露出了工作中存在的低效因素，因为你不得不使用两类分离的程序来完成一个同时需要连接和计算的大任务。

假设你想用 Microsoft Word 写一个建议，把它发送到一个 FTP 结点上，让其他人可以与你合作，然后，把它取回来最终成文，通过 e-mail 发送给第三方。自然地，你需要由启动 Word 来开始工作，用它来产生你的文档，然后在完成时保存到你的硬盘上。

接着，你必须启动独立的 FTP 程序（在必要时，还要拨号建立 Internet 连接），注册到 FTP 结点，从硬盘的目录中选择该 Word 文档，确认打开 FTP 二进制模式，提交文件。在合作者完成之后，你又要启动 FTP 程序，注册到 FTP 结点，改变目录，设置二进制模式，取回该文件，把它存到本地目录中。然后，你要退出 FTP 程序，启动 Word，打开文件，做进一步的改动，紧接着要保存文档。

最后，你需要启动独立的 e-mail 程序，产生一个消息，指定其地址，经过 e-mail 的连接过程后，把文件发送到它的最终目的地。当这一过程完成之后，你已经使用了三种完全不同的程序，每种都消耗一定的时间和内存，同时也带来出错的可能性。为什么软件会按这种方式使用呢？

为什么计算和连接还没有融合起来

计算和连接软件现在还未真正地融合起来，因为“它们总是这样”。然而这样的解释产生了“鸡和蛋”的问题。直到去年或前年，由于不存在实际要连接的地方或没有太大的需求去这样做，具有真正计算能力的商业应用（比如 Excel 和 Word）还不需要进行连接。计算和产生信息因此还是相当局部的事情；如果一个文件需要合作，那么往往是你与合作者工作于同一个办公室，可以通过交换软盘，或通过 LAN 来完成这项工作。

随着 Internet 的扩展，人们认识到即使是他们相隔很远，只要每个人都能够访问 Internet，两个（或更多）的人就可以在一定程度上协同工作；人们可以利用连接程序来彼

此进行通信。然而，由于存在着二类不同的程序，你不得不用连接程序进行通信，而用计算程序完成信息处理。

为什么计算和连接会很快地被融合起来

随着时间推移和计算机用户的变化，产生了一大批既需要创建信息又需要进行连接的用户，也产生了能够同时具备计算和连接功能的混和程序。实际上，现存的工具——Sun 的 Java 编程语言及 JavaScript——已经为像 Web 浏览器一样的连接程序增添了少许的智能。不幸的是，即使通过 Java applet，你的浏览器也只获得很少的智能和计算能力。这就是为什么 Java 程序是一种 applet 而不是应用；Java applet 也只能做到这一点。

为连接程序增加真正的计算能力，为计算程序增加真正的连接能力，并使其能够简单合理地实现，微软提出并完善了一种技术，这就是 ActiveX。通过使用 ActiveX 工具和方法（这将在第二部分“ActiveX 控制”中介绍），从无到有，或是从现存的程序出发，不仅能够建立一个同时具备计算和连接能力的程序，而且其开发时间也将相当的短。

第二章“微软的 Internet 未来蓝图”将会探索这些程序的可能性和潜力。