

请访问我们的Web站点寻求:

- 完整的工作实例
- 最新的文本
- 链接到附加的资源

- 用CGI和ISAPI编写网关
- 运行ISAPI过滤器扩展你的
IIS Web服务器的性能
- Java、Perl、C++和VBScript
- ActiveX控件



ActiveX Web Programming

ActiveX

Web开发指南

[美] Adam Blum 著

◆ 刘文智 孙波 吴健群 贾瑞新 译 ◆ 董涛飞 审校



WILEY



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

ActiveX Web Programming

ActiveX Web 开发指南

【美】 Adam Blum 著

刘文智 孙 波 吴健群 贾瑞新 译
董涛飞 审校

22

電子工業出版社

Publishing House of Electronics Industry

内 容 简 介

本书详细介绍了基于 Web 应用的 ActiveX 系列编程技术。全书的第一部分讨论了 Web 服务器端的编程,介绍了扩展 Web 服务器功能的 CGI 标准以及怎样用 C++ 进行 CGI 编程,同时特别提供了减轻 CGI 类处理负担的一个类,还介绍了用于扩展 IIS 功能的 ActiveX 框架及新的 ISAPI 标准及实例,并给出了 ISAPI 过滤器技术及实例。书中第二部分讨论了 ActiveX 客户端编程技术,介绍了 ActiveX 控件及制作交互式 Web 页的客户端标准。书中附有大量编程实例。本书是一本拥有许多关于 ActiveX 技术、大量实用源程序代码和可用类库的权威参考书。

本书对于想用基于 Web 的 ActiveX 新技术及其工具创造高级的、生动的 Web 页的计算机网络爱好者、网络开发者及网络编程人员极具参考价值。

Copyright © 1997 BY Adam Blum. All Rights Reserved. AUTHORIZED TRANSLATION OF THE EDITION PUBLISHED BY JOHN WILEY & SONS NEW YORK, CHICHESTER, BRISBANE, SINGAPORE AND TORONTO. NO part of this book may be reproduced in any form without the written permission of John Wiley & Sons Inc.

本书英文版由美国 John Wiley & Sons, Inc. 公司出版, John Wiley & Sons, Inc. 公司已将中文翻译出版权授予中国电子工业出版社。未经许可,不得以任何手段和形式复制或抄袭本书内容。版权所有,侵权必究。

书 名: ActiveX Web 开发指南

著 者: [美] Adam Blum

译 者: 刘文智 孙 波 吴健群 贾瑞新

审 校 者: 董涛飞

责任编辑: 李新社

特约编辑: 康宗朗

印 刷 者: 北京大中印刷厂

出版发行: 电子工业出版社、发行

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

URL: <http://www.phei.com.cn>

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 21.25 字数: 544 千字

版 次: 1998 年 5 月第 1 版 1998 年 5 月第 1 次印刷

书 号: ISBN 7-5053-4386-6
TP·2022

定 价: 35.00

著作权合同登记号 图字: 01-98-0097

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

目 录

绪言	(1)
第一章 程序员的 Internet 信息服务器	(4)
1.1 Internet 数据库链接器	(4)
1.1.1 准备使用 Internet 数据库链接器	(5)
1.1.2 编辑 Internet 数据库链接器文件	(6)
1.1.3 IDC 扩展的 HTML 标记	(7)
1.1.4 你能用 IDC 做什么	(8)
1.2 使用 IIS、IDC 和 SQL Server 的基于 Web 的定货系统	(8)
1.2.1 数据库定义	(8)
1.2.2 页的简介	(9)
第二章 网关和公共网关界面	(15)
2.1 为什么需要了解 CGI	(15)
2.2 什么是网关	(15)
2.3 CGI 标准	(16)
2.3.1 CGI URLs	(17)
2.3.2 输入	(17)
2.3.3 命令行参数	(18)
2.3.4 输出	(19)
2.3.5 环境变量	(21)
2.3.6 HTTP 请求标题	(23)
2.4 C++ 的 CGI 编辑	(23)
2.4.1 HTML Form 类	(23)
第三章 Perl 的网关编程	(28)
3.1 Perl 编程简介	(28)
3.1.1 Hello, Perl	(29)
3.1.2 数据类型	(29)
3.1.3 变量	(30)
3.1.4 特殊变量	(31)
3.1.5 操作符	(31)
3.1.6 函数	(32)
3.1.7 表格和数组函数	(33)
3.1.8 流程控制	(34)
3.1.9 输入和输出	(34)
3.1.10 模式匹配	(36)
3.1.11 子程序	(38)

3.2	Perl 作为 OLE 自动控件	(40)
3.3	Web 表单数据集的 Perl 文本	(41)
3.4	Perl 包用于 Web 编程	(42)
3.4.1	创建 HTML 表单	(42)
3.4.2	处理 HTML 表单	(44)
3.4.3	一张示例表单	(45)
第四章	ActiveX 服务器框架:ISAPI 的技术规范	(47)
4.1	CGI 存在的问题	(47)
4.2	ISAPI 的结构	(48)
4.3	ISAPI 编程界面	(49)
4.4	使用 ISA	(53)
第五章	用 Visual C++ 编写 Internet 服务器应用	(54)
5.1	将 CGI 的 C++ 程序改写为 ISAPI	(54)
5.1.1	使 pECB -> lpbData 中读取数据来代替标准输入	(54)
5.1.2	用 ECB 字段或 Get ServerVariable 替换环境变量	(54)
5.1.3	将所有的标准输出改写为 WriteClient	(54)
5.1.4	将状态返回为 ServerSupportFunction 调用	(55)
5.1.5	去掉 Main 函数,编写 HttpExtensionProc	(55)
5.2	MFC 对 ISAPI 的支持	(55)
5.2.1	ISAPI Extension 向导	(55)
5.2.2	MFC 的 ISAPI 类	(60)
5.3	ISAPI 编辑用到的其它类	(62)
5.3.1	为 ISAPI 开发扩展 HTMLForm 类	(62)
5.3.2	程序实例:SENDFORM ISA	(63)
5.4	OLEISAPI——在你的 Web 系统中集成 OLE 自动服务器	(67)
5.5	OLEISAPI 是如何工作的	(68)
第六章	ISAPI 过滤器	(70)
6.1	什么是过滤器	(70)
6.2	ISAPI 过滤器简介	(70)
6.3	建立 CVTDOC ISAPI 过滤器	(73)
6.3.1	CVTDOC 做什么	(73)
6.3.2	建立过滤器	(73)
6.3.3	建立转换程序	(78)
6.4	使用 CVTDOC	(81)
6.4.1	安装	(81)
6.4.2	转换程序	(81)
6.4.3	用法	(82)

6.5	MFC 对 ISAPI 过滤器的支持	(82)
6.5.1	ISAPI 过滤器向导	(83)
6.5.2	ISAPI 过滤器类	(84)
6.6	ACCSFILT;增强的 IIS 访问控制	(85)
6.6.1	问题	(86)
6.6.2	解决	(87)
6.6.3	设计	(87)
6.6.4	安装和配置	(87)
6.6.5	执行	(90)
第七章	使用 ActiveX 控制	(92)
7.1	ActiveX 组件库	(94)
7.1.1	新项目	(94)
7.1.2	图表	(94)
7.1.3	动画按钮	(98)
7.2	表单控件和 HTML 布局控件	(99)
7.3	ActiveX 控件便笺	(101)
第八章	编写 ActiveX 控件	(106)
8.1	ActiveX 控件框架	(107)
8.1.1	WebImage 控件实例	(109)
8.1.2	ActiveX 模板库	(114)
第九章	VBScript	(120)
9.1	VBScript 的性能	(120)
9.2	在 HTML 中集成 VBScript	(121)
9.2.1	在 HTML 中尝试 VBScript	(122)
9.2.2	在 HTML 表单中使用 VBScript	(125)
9.3	和对象一起使用 VBScript	(129)
第十章	ActiveX 服务器	(132)
10.1	ActiveX 服务器对象	(133)
10.1.1	Request 对象	(133)
10.1.2	Response 对象	(135)
10.1.3	Session 对象	(136)
10.2	ActiveX 服务器组件	(138)
10.2.1	Connection 组件	(138)
10.2.2	Browser Capabilities 组件	(140)
10.3	购货篮实例	(142)
10.3.1	购货区	(146)

10.3.2 购货篮操作	(146)
10.3.3 数据库定义	(148)
第十一章 Java、Visual J++ 和 ActiveX	(151)
11.1 Visual J++ 开发环境	(151)
11.1.1 运行 Applet Wizard	(152)
11.1.2 利用 Visual J++ 设计 Java applet 的用户界面	(158)
11.2 用 VBScript 控制 Java applets	(172)
11.3 用 Java 来控制 COM 对象	(173)
11.3.1 使用 Type Library Wizard 产生类	(185)
11.3.2 使用 Java 的 OLE 控件	(190)
11.3.3 Java 的安全性和 COM 的使用	(191)
附录 A 源代码	(193)
A.1 SENDFORM	(193)
A.1.1 MAKEFILE	(193)
A.1.2 SENDFORM.H	(193)
A.1.3 SENDFORM.CPP	(196)
A.1.4 HTMLFORM.H	(202)
A.1.5 HTMLFORM.CPP	(205)
A.1.6 ASSOCARR.H	(212)
A.2 CVTDOC	(214)
A.2.1 MAKEFILE	(214)
A.2.2 CVTDOC.DEF	(214)
A.2.3 CVTDOC.CPP	(215)
A.3 ACCSFILT	(220)
A.3.1 MAKEFILE	(220)
A.3.2 ACCSFILT.DEF	(221)
A.3.3 ACCFILT.H	(221)
A.3.4 DIVECALC	(226)
A.4 OPTIONS	(229)
A.4.1 OPTIONS.JAVA	(229)
A.4.2 IDD_OPTIONS.JAVA	(234)
A.4.3 DIALOGLAYOUT.JAVA	(237)
A.4.4 OPTIONS.HTML	(241)
A.5 OLEMSG	(241)
A.5.1 OLMMSG.AJVA	(241)
A.5.2 IDD_SENDMSG.JAVA	(246)
A.5.3 OLEMSG.HTML	(248)
A.6 SHOPBSKT	(249)

A.6.1	PRODLIST.ASP	(252)
A.6.2	ADDPROD.HTM	(252)
A.6.3	ADDPROD.ASP	(253)
附录 B	ISAPI 技术规范 and 头文件	(254)
B.1	HTTPEXT.H	(254)
B.2	HTTPFILT.H	(257)
B.3	AFXISAP.H	(266)
附录 C	ActiveX 服务器对象参考	(280)
附录 D	微软的 COM Java 类	(281)
D.1	applet	(281)
D.2	awt	(301)
D.2.1	MentX.java	(301)
D.2.2	MenuXConstants.java	(322)
D.2.3	CaretX.java	(327)
D.2.4	ByteArrayImageSource.java	(328)

绪 言

什么是 ActiveX?

最近,微软提出了 ActiveX 系列技术,以便 Internet 的开发者们能建立更高级的,更好的基于 Web 的应用。ActiveX 以服务器框架的形式提供服务器方技术:Internet 服务应用程序界面 (ISAPI)的技术规范和 ISAPI 过滤器的技术规范。ActiveX 还包含各种客户方技术,其中很重要的,在本书中将特别提及的一点是可在 Web 页或 Visual Basic 应用程序、Active Scripts(一种在 VBScript 标准中嵌入客户方功能的 ActiveX 控件),用它可在 Web 浏览器、Java applets,尤其是那些用微软新一代 Java 开发环境(现在称作 Jakarta)开发的软件中提供客户方表达功能。

ActiveX 的 Web 编程包含了从网关编程到公共网关界面(CGI)和微软新的 ISAPI 标准。程序实例将采用 C++ 和 perl 语言。本书涉及使用 OLE 控件和 VBScript 实现服务器或网关应用和客户方开发的方法。随着 ActiveX 服务器和客户方新技术的涌现,一个特定的功能集会经常以 ISAPI 网关、OLE 控件、VBScripts 或 Java applets 的形式提供给用户。绝大部分应用将从可使用的编程技术的组合中受益。实际上,我将通过实例来说明采用混合编程技术比采用单一技术能生成更好的 Web 应用。

Web 发展历史

汇集静态 HTML 内容建立 Web 站点已经成为一种平常的技术。现在,可以通过交互程度找出最好的 Web 站点。这种交互是指允许用户对站上的内容作出反应,传送他或她自己的反馈信息,寻找想要的信息,或者仅只是浏览信息。这是一种比静态 HTML 页更为生动的功能。它需要 Web 网关或客户应用程序(此程序运行于用户的 Web 浏览器)的开发者参与。更明确地说,这需要建立的 Web 应用将这些技术完美地组合在一起。

第一代建立 Web 服务器方功能的方法是 CGI 程序。尽管 CGI 是本书涉及的一个很重要的标准,但它有很多缺陷。不过在 ISAPI(ActiveX 服务器框架的一部分)中已经被消除了。由于引入了 OLE 控件和 ActiveX Scripts,即使功能强大的 ISAPI 网关运行于极限功能状态,通过用户的工作站也能对其进行最好的操作。

本书的目的

如果你拥有的一切是一把锤子的话,整个世界看起来就象一颗钉子。如果那把锤子是服务器网关编程,那么你可以用网关来建立你的应用,大多数应用都可通过这种方式建立起来。但这并不总是正确的方法。由服务器来完成所有的工作可能会在客户和服务器之间形成不必

要的数据流。如果用户需要操作一个本地可用的数据,他应该能得到本地应答,而不是通过 CGI 网关让服务器来完成这项工作。如果那锤子是客户方文本描述(scripting)的话,你可以编写应用程序,让客户机完成所有的计算和数据传送工作。但这并不是建立应用的最好办法。在本书中我将提供可用于 Web 开发的所有 ActiveX 新技术,以便读者能选择正确的工具,或者说正确的工具组合,并用这些工具有效地创造出高级的、生动的 Web 应用。

在微软的 ActiveX 软件开发包(SDK)中引入了各种各样的技术,所有的这些技术都是新的。但每种技术可应用于何种场合并未完全说明。从这一点来看,关于如何组合这些技术以制作基于 Web 的复杂应用的指导和知识几乎不存在。本书将使用一些示例应用来说明每项技术的使用以及何时使用某一技术最有效。大一些的应用通常是一个技术的混合体,用以说明如何利用多种工具创建 Web 应用。我们将演示的一个较大的应用是基于 Web 的在线仓库,它使用 ISAPI 网关(可用 C++ 类库按任意的网关协议对其进行设置)来显示产品和进行订购。在客户方,使用 ActiveX 控件来操作这些数据,在按预定格式将这些数据发送出去之前用 VBScript 来完成客户方校验。习惯上,这项工作是由网关来完成的,但是考虑到可能出现的数据错误,如果还这样做就必须承担使用冗余装置所带来的费用。在另一个例子中,客户浏览器用 ActiveX 控件将数据用图表的方式表示出来。不必将整个图表传送给用户,只要传送数据并允许控件接收和显示数据即可。这样会省去大量的图象下载,并使按图形方式显示数据的页上载很快。采用象 ActiveX 控件(控件 COM 标准的一种简化)这样的技术没有理由不在客户浏览器中完成计算工作,只要数据已存在于工作站中。最后,Java applets 可以做更多类似的客户方工作,并且很容易下载,在客户浏览器中可运行如飞。微软的 Java VM 工具独到之处在于能够充分发挥已有的 COM 对象的功能。实际上,Java 的原型比 C++ 更接近 COM 原型。所以在这样一本关于 ActiveX 的书中讨论微软 Java 工具是非常适合的。

本书的一个重要目的是通过包含尽可能丰富的内容而成为一本参考书,并且提供在其它任何 ActiveX SDK 参考资料中找不到的内容和应用。这些资料如:SDK、Visual C++/MFC 文档中关于 ActiveX 的部分(如:VC 增强版中辅助 ISAPI 编程)或 Web 站点 www.microsoft.com/intdev 上的内容。这些都是本书反复参考的、重要的资料来源。但是贯穿全书的示例程序所使用的每一项技术都是有用的,并且在其它地方不可能找得到。书中同时提供给读者的还有编程工具,比如用于协助生成示例程序的 C++ 类库。书中许多这样的工具仅为本书对各种技术提供实例。其它经过严格考验的解决实际客户问题的程序可供 Web 开发者参考。现在,本书提及的每一个工具几乎都在 Web 站点或其它场合大量使用。

作为一本参考书,本书试图为每一个重要的、新的 ActiveX 规范和技术提供足够的信息。并尝试彻底说明使用每一个 API 的理由。书中的示例程序采用的每一项技术都做了有用的、富有成效的工作,但并不打算结合实例展现大量的 HTML 页或完全的工作站点。在正文和本书的 Web 站点中用大量的 HTML 结合示例程序构成更为成熟的例子只会使每个程序的目的和结构变得模糊不清。我提及这点,稍微有点免受责难的意味。因为在我的 Web 开发研讨班上,经常会有精通 Web 的人悄悄地问:“HTML 在哪里?”答案通常是:“哦……它会由这个 Internet 服务器应用程序或者 ISAPI 过滤器或者 perl script 或者别的什么东西生成。”如果你看到到处都是静态 HTML 的话,你会明白本书不会达到它的即定目标:一本拥有许多关于 ActiveX 的技术、大量有用的实例源代码、可重用类库和实用程序的权威的编程参考书。

本书结构

本节第一部分论述 Web 服务器方编程。首先,我简要介绍了微软新的 Internet 信息服务器(IIS)的功能,其中特别提及 IIS Internet 数据库联接器,它允许不必编程即可建立小的应用,如收集数据存入数据库或从数据库生成报告。其次介绍了扩展 Web 服务器功能的 CGI 标准(即使引入了 ActiveX 和 ISAPI,知道这些还是很重要的),以及如何用 C++ 进行 CGI 编程。同时提供了一个类(本书特有)以减轻 CGI 类型处理的负担。下一章涉及到如何用 Perl 建立可用于 CGI 和 ISAPI 的网关。随后一章论述扩展 IIS 功能的 ActiveX 框架以及新的 ISAPI 标准和几个使用 ISAPI 界面的实例,包括一个协助建立 ISAPI 程序的 C++ 类(本书另一个特有的工具)。这部分最后一章是关于 ISAPI 过滤器技术规范 and 实例。这些实例说明如何让 Web 服务器的性能获得戏剧性扩张。在此提及的所有 ISAPI 实例是专为本书编写的。作为增强的 IIS 访问控件的 ISAPI 过滤器也只能在本书中找到。

本书另一部分讨论了 ActiveX 引入的 Web 客户方技术。最主要的部分是 ActiveX 控件,它是一种生成很容易分布于 Internet 上的 OLE 控件的标准。此部分也涉及到一种制作交互式 Web 页的客户方描述标准。这种标准称为 ActiveX Scripts,其主要的实例是 VBScript。

第一章

程序员的 Internet 信息服务器

微软新的 Web 服务器, Internet 信息服务器(IIS)和基本的 HTTP 服务器相比, 有三个主要特点: Internet 数据库链接器(IDC), 能直接利用 ODBC 数据库; 与 Windows NT 安全集成; 稳健的、高性能的可与网关程序通信的 ISAPI。IIS 是一个非常强大的产品, 具有许多值得推荐的性能, 但上述这些是与其它产品的真正的差别。从开发者的角度来看, IIS 的特殊意义是什么呢? 显然, 很大部分来自 ISAPI(也称作 ActiveX 服务器框架)。是它打开了建立稳健的、高性能、易于确定规模的、成熟的 Web 应用之门, 并使之与 IIS Web 服务器紧密相连。本书将主要讨论这一特点, 并将大量的注意力集中于开发基于 ISAPI 的应用所需的技术上。(这些应用也称为 Internet 服务应用 ISAs)

但是, 在将大量篇幅投入到对 ISAPI 的描述之前, 我想讨论一下 IDC。虽然这个部件是为了供非程序员的 Web 使用者操作数据库而开发的, 但对 Web 开发者而言, 仍是一个十分有用的工具。IDC 可用于在已有的基于 SQL 的系统中增加简易的特定的报告功能和从 Web 页上查询的功能。通过 HTML 的形式收集数据, 并以记录方式存放到 ODBC 数据库的 Web 页可以只用极少的工作就建立起来, 且与传统的编程方法截然不用。这些 Web 页可用任意的 HTML 编辑工具生成, 并只需加入一小点内容以鉴别数据库字段名。这个过程如此的容易和直观, 以至于很难想象这就是编程。实际上, 也许不能再用原有的思想来考虑这个问题。不过, 用这种方法可以取代许多用传统开发方法必须进行的工作。所以真正的 ActiveX Web 开发者必须搞清楚怎样最有效地使用 IDC。

1.1 Internet 数据库链接器

实际上, IDC 是一个 ISAPI 动态链接库的实例。是用于从 IIS HTTP 服务器访问 ODBC 数据源的 Internet 服务器应用。这个特殊的动态链接库即 httpodbc.dll, IDC 的结构如图 1.1 所示。

无论何时, IIS 遇到扩展名为 .IDC 的文件时, 都会自动引用 Internet 数据库链接器。这是怎样实现的呢? IIS 载有几个脚本映射。它们位于注册关键字 HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlset/Services/W3SVC/Parameters/Script Map 下。

图 1.2 展示了 IIS 装载的脚本映射: 你可在其中加入相应的内容, 使 IIS 遇到 .pl 的文件时运行 Perl.exe。

为了让 Internet 数据库链接器开始运行, 你可以创建一个扩展名为 .IDC 的 IDC 文件, 其中包含必需的 SQL 语句, 用于在你的数据库中插入一个记录(通常是一条 Insert 语句), 或从数据库中检索一张记录表或一条特殊记录(通常采用 Select 语句)。

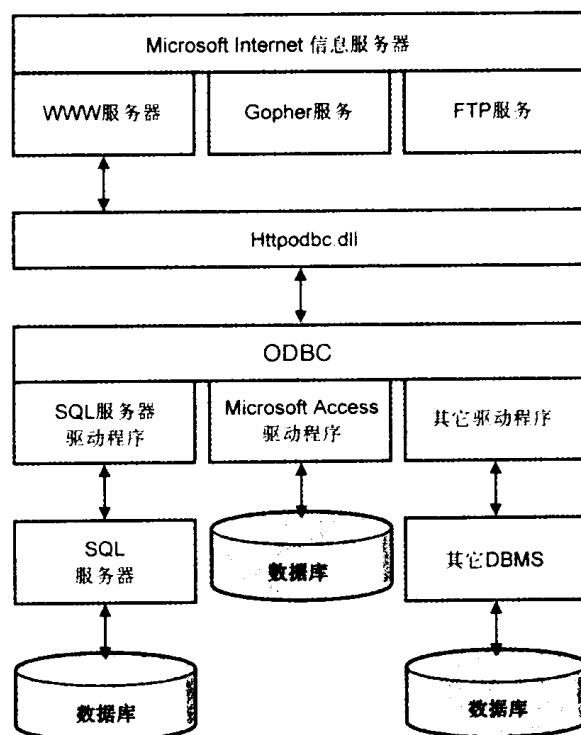


图 1.1 Internet 数据库链接器结构

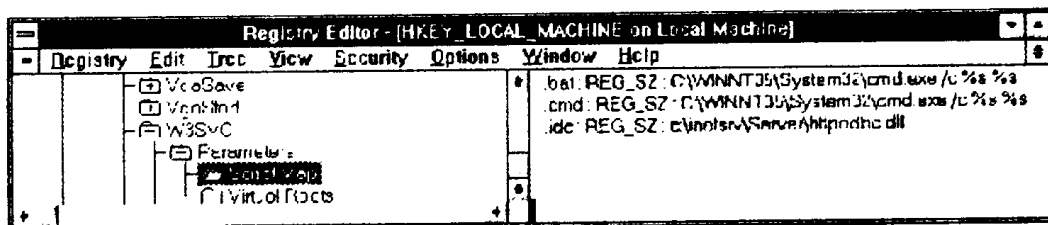


图 1.2 脚本映射的注册关键字

1.1.1 准备使用 Internet 数据库链接器

要使用 IDC, 在你的 IIS 服务器中必须装有 ODBC。这是安装 IIS 时的一条选项, 希望你选择了它, 以便能马上开始工作。否则, 你必须重新运行 IIS 的安装程序完成安装。为了运行 IDC, 第一步在确保 ODBC 已经存在的前提下, 创建一个系统数据源。为此在控制面板中单击 ODBC applet, 然后单击 System DSN 按钮。如果你运行的是 SQL Server 6.5, 则一个 System DSN 已经存在, 叫做 Local Server。当然, ODBC 进行的工作是以其它数据库为背景的, 比如 Microsoft Access 和 RDBMS。如果你可选择 SQL Server 或 Access 进行工作的话, 考虑到所能提供的高水平的处理, 我建议你使用 SQL Server。在本书中, 几乎所有的例子都使用 SQL Server。如果你计划一直使用 IDC 同某个特定的数据库通信的话, 你可能希望设置这个特定数据库的数据库名。图 1.3 表示将 Local Server DSN 改为将一直使用的 Pubs 数据库

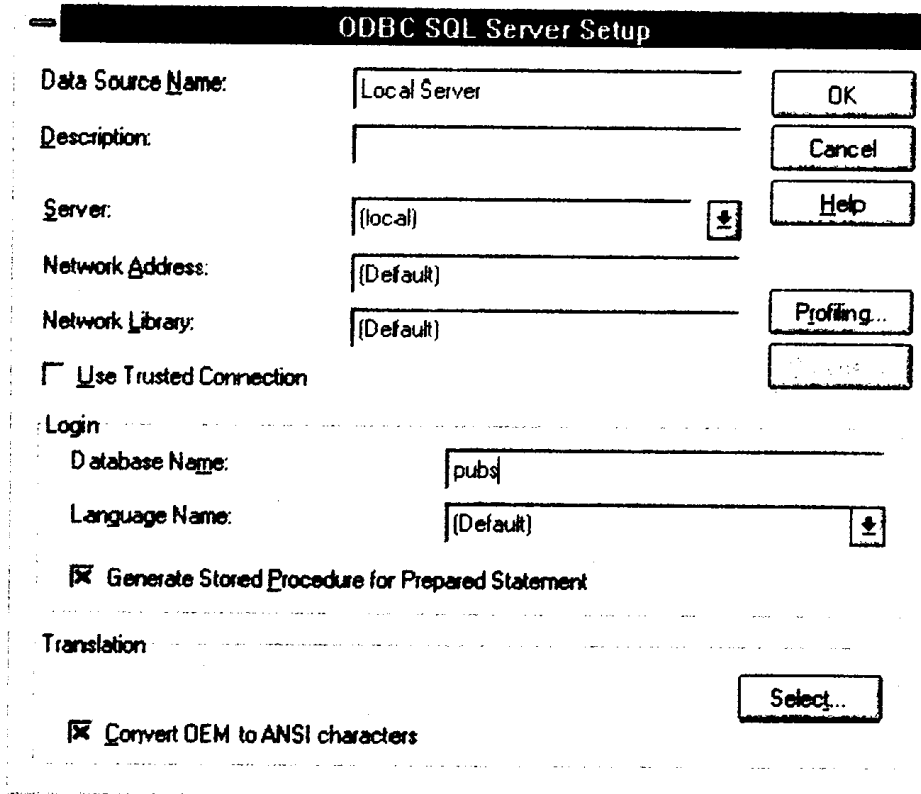


图 1.3 配置 ODBC 数据源

1.1.2 编辑 Internet 数据库链接器文件

一旦配置完毕,你就可以开始创建 Internet 数据库链接器文件了。这包括写一个 .idc 文件,以决定向数据库发送什么命令,同时需要建立一个 HTML 模板文件用于控制 Internet 数据库链接器返回数据的显示。

以下是一个 IDC 文件例子,名叫 Sample.idc,可在 IIS 中找到。它进行的工作需要用到微软的 SQL Server 示例数据库 Pubs。

```
Datasource;Local Server
Username;sa
Template;sample.hix
SQLStatement;
+ SELECT au_lname,ytd_sales from pubs.dbo.titleview where
+ ytd_sales>5000
```

注意 Datasource; 字段指出一个已经用 ODBC applet 设置的 System DSN, Username; 是一个有足够优先级的用户名,以便能完成一条 SQL 语句。你不必将 Sa 作为用户名。虽然 IIS 中的示例程序使用的是 Sa,但你有更多的选择。Template; 名是一个含有 HTML 供指示和嵌入字段代码以决定从服务器返回的数据格式的文件。按照惯例,其扩展名为 .hix(大概是表示 HTML)。但这并不是必需的。出于管理方便考虑,当这样的模板文件用于 .IDC 文件时,我们

建议你使用同样的扩展名。文件中最后一个字段是 SQL 语句组。其内容是送到后台数据库的完整的 SQL 语句。注意在每行开始都有一个加号。我们极力建议你采用微软的 SQL Server 中 ISQL 工具或 Access 的 query builder 来交互式地开发和测试 SQL 语句,然后粘贴到此文件中并在每行前加上一个加号。在 .idc 文件中用到的 SQL 语句通常是用于记录列表的 SELECT 语句或加入记录的 INSERT 语句。一条带有 WHERE 从句的 SELECT 语句可检索一条特定的记录,而一条 UPDATE 语句可修改一条记录。

现在,我们要建立一个 HTML 模板文件以接收数据库返回的结果。以下是显示运行刚才的 IDC 文件后产生的 Sample.htx 文件。

```
<HTML>
<HEAD><TITLE>Authors and YTD Sales</TITLE></HEAD>
<TABLE BORDER>
<B><TR><TH>Author<TH>YTD Sales</B>
<%begindetail%>
<TR><TD><%au_lname%><TD>$ <%ytd_sales%>
<%enddetail%>
</TABLE>
</BODY>
</HTML>
```

这看上去似乎与任何普通的 HTML 页没什么不同,不过请看第五、第六和第七行。它们带有一些<%...%>标识,这与一般的 HTML 页不同。它们是 HTML 的 IDC 扩展,决定了从数据库返回的数据将被放到 HTML 页上的什么位置。标记<%begindetail%>和<%enddetail%>允许你处理从数据库系统返回未知数量的行时的情况。每行均被这些标记之间的 HTML 和嵌入字段名规定了格式。处理这种问题常用的方法是将返回数据放到一个 HTML 表中再以此行的方式显示出来。但这并不是必需的。你可以将所有的字段代码放于一行,并用段落标记(<p>)来逐行浏览。每行的字段数据用标记<%fieldname%>进行检索。一般,在 IDC 中,比如本例,每条 SELECT 语句提到的字段都是字段代码。要注意的是,虽然与 IIS 中的 Sample.idc 很相似,但本例却是相当简单的。这样相信会使你更易于理解和使用。

1.1.3 IDC 扩展的 HTML 标记

完整的 IDC 文档可从 IIS 帮助文件中找到。作为快速参考,并考虑到在帮助文件中并未将其设计为易于阅读的格式,所以我们将一些常用的标识和变量列于表 1.1。在你编程 HTML 模板文件(.htx)时,对你可能有用。

表 1.1 HTX 标识和变量

标识	描述
< %begindetail% > < %enddetail% > < %if% > .. < %else% > < %endif% >	HTML 扩展文件中用关键字 < %begindetail% > 和 < %enddetail% > 围住的部分用于合并来自数据库的数据。 有条件地显示某一 HTML 集。例如： < %if CurrentRecord Eq 0 % > No records returned! < P > < %endif% >
CurrentRecord	用作 < %if% > 标识的参数。
MaxRecord	IDC 能返回的记录数的最大值。 在 IDC 文件中这是一个可选项。如果没有设置,则实质上为无穷大(40 亿)。
EQ	在 < %if% > 语句中作为条件测试。如果变量 1 等于变量 2 则为“真”。
LT	另一个条件测试。如果变量 1 小于变量 2 则为“真”。
GT	如果变量 1 大于变量 2 则为“真”。
CONTAINS	如果变量 1 的任何部分包含了字符串变量 2 则为“真”。

1.1.4 你能用 IDC 来做什么？

即便是超出了基于 Web 的应用的范围,Internet 数据库链接器也是很有用的。为什么呢?因为它提供了一个非编程的,非常易于维护的与数据库中数据相联系的接口。维护这样一个系统(例如修改一个报告页)唯一需要的技术是 HTML 编辑及了解正在处理的表的字段名。通常说来,在运行正常的情况下,甚至不必修改 SQL 语句。从最终用户层维护系统所需的技术最少的角度而言,这是一个很理想的系统。

如果你要让 Web 浏览器访问已有的、由其它几个应用提供的数据库时,IDC 特别有用。由于以 HTML 形式提供的数据库入口使得 IDC 难以提供复杂的(或者说任何的)校验逻辑,所以 IDC 在报告和查询方面比作为数据库入口更为有用。对于某些以简化简单数据收集为目的的应用来说,将 IDC 作为数据库入口是可以接受的,而对其它的大多数的应用而言,则是行不通的。幸而,对这种限制尚有一线希望:客户方校验。这意味着将校验逻辑放于客户方浏览器以检查每个字段名。这样,记录的入口更有可能成功。在随后的章节中,我们将展示如何利用 VBScript,通常为 Active Script 来简化这一操作。

为给你提供一个这样的例子,同时作为你自己系统的基础,我们将向你展示一个用 Internet 数据库链接器建立的简单的定货输入系统。

1.2 使用 IIS、IDC 和 SQL Server 的基于 Web 的定货系统

使用这一技术的典型例子是基于 Web 的产品定货系统。在此,我们将介绍这个例子,并且仅用 Internet 数据库链接器就能使它工作(根本不用编程!)在本书结束的时候,我们将给它加入其它的功能,以演示使用 ActiveX 时可用的各种编程技术。

1.2.1 数据库定义

最初的 Web 定货系统将以两个主表为基础:一个产品表和一个定货表。图 1.4 指出了产

品表在 Microsoft SQL Server 6.5 Enterprise Manager 中的定义。注意即然 IDC 可与任何 ODBC 数据源交谈,你可在你选择的数据库中复制这个数据库的定义。

Key	Identity	Column Name	Datatype	Size	Nulls	Default
		prodno	int	4		
		product	char	255	✓	
		description	text	16	✓	
		price	money	8	✓	
		discount	real	4	✓	
		available	int	4	✓	
		categoryno	int	4	✓	
		manufacturno	int	4	✓	
		imagefile	char	255	✓	

图 1.4 产品表

关键字段 prodno 用于唯一地标识产品。产品的文本名存放于 product 字段。这些由非程序员级 Web 使用者/管理者维护的、类似于 HTML 模板中使用的字段名被设计得尽可能的简单易读。因此,price 指产品的购买价格,Available 指出可供出售的产品数量。Discount,如果有的话,标出了产品的降价百分比。它通过全面更新此列的表使销售更容易完成。随后的两个字段供以后增强系统功能用。categories 表和 manufacturer 表用于提供检索。imagefile 字段提供了一个存贮产品图片的 GIF 或 JPEG 文件所在路径名。

图 1.5 显示了一个定货表的定义。

orderno 字段是特有的关键字,prodno 字段用于在产品表中检索。price 为产品的单价,所付价钱由 $\text{price} * \text{quantity}$ 得出。paymethod 字段是一个供在付款方式表中进行检索的数字,例如,0 表示现金支付,1 为支票...,接下来的五个字段指出了发货地址。这并不能从保留的客户表中得到,因为我们假定货物能发往任意的地址。同时,也有顾客 ID 号字段和顾客名字段,这使得数据库有些不正规,但对我们的这个简单例子比较有用。随着我们书的进展,这张表会变得更为复杂,但是这些已经可以作为我们随后的例子使用的基础了。

1.2.2 页的简介

现在我们要将一些 HTML 页放到一起,以使用户可从产品表中列出的产品中选取可用的物品,或者将定购的产品放入定货表中。在此我们用顶层页来引用 Products.idc 文件,显示所